



**UNIFACS**

UNIVERSIDADE SALVADOR

LAUREATE INTERNATIONAL UNIVERSITIES'

**UNIFACS UNIVERSIDADE SALVADOR  
PROGRAMA DE PÓS-GRADUAÇÃO  
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO**

**ADALBERTO ALVES ABRAÃO**

**ASSISTENTE FINANCEIRO VIRTUAL BASEADO EM PROCESSADOR DE  
LINGUAGEM NATURAL, RECONHECIMENTO DE VOZ E XBRL**

Salvador  
2016

**ADALBERTO ALVES ABRAÃO**

**ASSISTENTE FINANCEIRO VIRTUAL BASEADO EM PROCESSADOR DE  
LINGUAGEM NATURAL, RECONHECIMENTO DE VOZ E XBRL**

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação da UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Paulo Caetano da Silva.

Salvador  
2016

## FICHA CATALOGRÁFICA

Elaborada pelo Sistema de Bibliotecas da UNIFACS Universidade Salvador, Laureate International Universities)

Abraão, Adalberto Alves

Assistente financeiro virtual baseado em processador de linguagem natural, reconhecimento de voz e XBRL./ Adalberto Alves Abraão.- Salvador: UNIFACS, 2016.

145 f. : il.

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação de UNIFACS Universidade Salvador, Laureate International Universities como requisito parcial à obtenção do título de Mestre.

Orientador: Prof. Dr. Paulo Caetano da Silva.

1. XBRL. 2. Assistência financeira virtual. 3. Taxonomias financeiras. I. Silva, Paulo Caetano da, orient. II. Título.

CDD: 005.133

Dedico este trabalho à meu querido e amado filho Renan, a meus pais, por me conscientizarem da importância da busca do conhecimento, a minha amada esposa, que está sempre ao meu lado em todos os momentos.

## **AGRADECIMENTOS**

Agradeço à minha esposa, Neire Abraão, companheira e amiga de todas as horas, pelo apoio e compreensão às minhas eventuais ausências do convívio familiar para a elaboração desse trabalho.

Ao meu filho, que por vezes, reclamava que o “mestrado” estava impedindo que eu brincasse com ele, e abria um sorriso revigorante quando eu interrompia por alguns minutos a elaboração dessa dissertação para me dedicar integralmente à ele.

A minha mãe, Maria Celeste, pelo grande incentivo e exemplo de dedicação aos estudos.

A empresa Petrobras que acreditou na minha capacidade e me deu a oportunidade de realizar esse desafio.

Ao professor Doutor Paulo Caetano, que me orientou a desenvolver esse trabalho com seriedade e competência.

## RESUMO

Para investir em uma empresa é importante obter informações sobre a situação financeira atual e passada. Extrair essa informação de um sistema de informação que utiliza taxonomia reconhecida pelo mercado financeiro, utilizando linguagem natural, é uma facilidade para o investidor que não tem conhecimento e nem expertise na área de informática. Para auxiliar pessoas em determinada área de conhecimento sem exigir delas conhecimento prévio em Tecnologia da Informação, surgem os softwares de assistência virtual como uma alternativa. Contudo, observou-se que não há disponível um sistema de informação que interaja com o usuário através de linguagem natural para responder questões financeiras sobre empresas baseado em informações disponíveis em relatórios eletrônicos. Este trabalho apresenta uma ferramenta de assistência virtual que mantém um diálogo com o usuário utilizando linguagem natural, em inglês ou português, com a finalidade de responder questões financeiras sobre empresas que atuam no mercado financeiro americano. Essa ferramenta, nomeada de Assistente Financeiro Virtual, reconhece as perguntas do usuário através de processamento de linguagem natural e fornece as respostas baseada em informações disponíveis em demonstrativos financeiros eletrônicos em tecnologia XBRL, conformes com a taxonomia US-GAAP. A arquitetura, baseado em serviços, do Assistente, a sua implementação, a seleção e configuração do serviço de processamento de linguagem natural, também são apresentadas. A partir dos testes e avaliações da implementação do Assistente constatou-se que as perguntas do usuário foram respondidas corretamente e os resultados de desempenho foram compatíveis com os softwares de assistência virtual de propósito geral disponíveis no mercado. Este trabalho contribui com a área de software de assistência virtual e com área de extração de informações em formulários eletrônicos, apresentando uma arquitetura para assistentes de propósito específico baseada em serviços e processamento de linguagem natural.

**Palavras-chave:** Assistência financeira virtual. XBRL. Taxonomias financeiras. Arquitetura para assistentes virtuais de domínio restrito. Assistente de propósito específico. Linguagem natural.

## ABSTRACT

When an investor want to invest in a company, it is necessary to know the current and past financial health of this company. Extracting this financial information from an information system that uses recognized taxonomy by the financial markets, using natural language, is a facility for the investor who has no knowledge and no expertise in computer science. As alternative to assist people in a particular area of knowledge without requiring them prior knowledge in information technology, there are the virtual assistance software. However, it was observed that there is not available a software of virtual assistance that answers made user questions in natural language about financial health of companies based on available information in electronic reports. This paper presents a virtual assistance tool to maintain a dialogue with the user using natural language, in English or Portuguese, to answer financial questions about the companies that acts in the US financial market. This tool, named Financial Virtual Assistant, recognizes the user questions through natural language processor and provides responses based on available information in electronic financial statements in XBRL technology, in accordance with US-GAAP taxonomy. The Assistant implementation, its architecture, based on services, the selection and configuration of the used natural language processor service, are also presented. As a result of the tests and evaluations of the assistant implementation, it was observed that the implemented Financial Virtual Assistant acted as proposed, and correctly answered user questions, and presented compatible performance results with the available virtual assistant software of the general purpose. This work contributes to the field of virtual assistance software and information extraction area of electronic forms, with an architecture for the closed domain assistants, based in services and natural language processing.

**Keywords:** Financial Virtual Assistance. XBRL. Financial taxonomies. Architecture for closed domain virtual assistants. Assistant for specific purpose. Natural language.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas da pesquisa científica .....	23
Figura 2 - Arquitetura de Assistente pessoal com interação por voz baseado em agentes .....	32
Figura 5 - Arquitetura do Framework baseado em Natural Language Understander	36
Figura 6 - Arquitetura Orientada a Serviços do SAP .....	37
Figura 7 - Composição dos elementos da implementação da arquitetura de referência .....	38
Figura 8 - API que suporta a Assistant.ai .....	40
Figura 9 – Siri – Tecnologia para integrar diversos serviços .....	42
Figura 10 - Arquitetura em camadas do Assistente Financeiro Virtual.....	46
Figura 11 - Arquitetura para assistentes virtuais de propósito específico. ....	47
Figura 12 - Arquitetura do Assistente Financeiro Virtual .....	50
Figura 13 - Ciclo de vida de uma transação típica controlada pelo Coordinator. ....	52
Figura 14 - Diagrama de Sequência de uma consulta ao Assistente Financeiro Virtual.....	61
Figura 15 - Telas da aplicação móvel do Assistente Financeiro Virtual .....	66
Figura 16 – Telas da aplicação do Assistente Financeiro Virtual em HTML/Javascript .....	67
Figura 17 - Telas em inglês da versão móvel do Assistente Financeiro Virtual .....	68
Figura 18 – Telas em inglês da versão HTML / Javascript do Assistente Financeiro Virtual.....	69
Figura 19 - Diagrama de classes da implementação da camada Orchestration .....	71
Figura 20 - Classe Coordinator .....	72
Figura 21 - Classe AssistantContext .....	73
Figura 22 - Classe RequestParameters .....	75
Figura 23 - Diagrama das classes NaturalLanguageProcessor e NLPServiceAdapter .....	76
Figura 24 - Diagrama de classes do objeto de resposta do NLP da Api.ai .....	78
Figura 25 - Classe AssistantResponse .....	79
Figura 26 – Diagrama de classes relacionadas a classe DialogKnowledgeManager	80
Figura 27 - Classe ServiceResponse .....	83

Figura 28 - Diagrama de classes relacionadas com a classe FinancialInformationServicesSwitch.....	101
Figura 29 - Diagrama de classes da fábrica de serviços financeiros.....	103
Figura 30 - Classe abstrata FinancialInformationService .....	104
Figura 31 - Diagrama das classes Financial Service implementadas .....	106
Figura 32 - Classe CompanyData e FinancialConcept.....	108
Figura 33 - Classe XbrlUsgaapWebService .....	110
Figura 34 - Diagrama de Sequência de uma transação bem sucedida.....	114
Figura 35 - Resultados do Assistente Virtual Siri da Apple .....	122
Figura 36 – Sequência de respostas do Siri para perguntas curtas incompletas ....	123
Figura 37 – Perguntas que o Siri identificou os parâmetros financeiros.....	125

## LISTA DE QUADROS

Quadro 1 - <i>Strings</i> de busca utilizadas nas respectivas bibliotecas digitais .....	24
Quadro 2 - Avaliação das plataformas NLP .....	85
Quadro 3 - Entity CommandExpressions .....	89
Quadro 4 - Fragmento da entity Company .....	90
Quadro 5 - Fragmento da entity USGAAP_BalanceSheet .....	92
Quadro 6 - Fragmento da entity USGAAP_BalancoFinanceiro .....	93
Quadro 7 - Parte da entity YearPeriod .....	94
Quadro 8 - Parte da Intent “What is the Financial Concept of Company At Period” ..	95
Quadro 9 - Lista das Intents da versão em inglês .....	97
Quadro 10 - Lista das Intents da versão em português.....	97
Quadro 11 - Nome da intent e o respectivo valor do parâmetro action .....	99
Quadro 12 - Considerações sobre os requisitos do Assistente Financeiro Virtual ..	126

## **LISTA DE TABELAS**

Tabela 1 - Tempos de respostas do Assistente Financeiro Virtual.....	120
--	-----

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>15</b>
1.1 CONTEXTUALIZAÇÃO	15
1.2 JUSTIFICATIVA	17
1.3 MOTIVAÇÃO	19
1.4 OBJETIVO	20
1.5 METODOLOGIA	21
1.6 ORGANIZAÇÃO DA DISSERTAÇÃO	25
<b>2 ASSISTENTES VIRTUAIS</b>	<b>28</b>
2.1 FRAMEWORK PARA INTERAÇÃO POR VOZ	29
2.2 FRAMEWORK PARA ASSISTENTE MÓVEL SENSÍVEL AO CONTEXTO	30
2.3 ASSISTENTE VIRTUAL COM INTERAÇÃO POR VOZ BASEADO EM AGENTES	31
2.4 FRAMEWORK PARA CONSTRUÇÃO DE ASSISTENTES VIRTUAIS DE DOMÍNIO RESTRITO	33
2.5 SOFTWARE DE ASSISTÊNCIA PESSOAL ORIENTADO A SERVIÇO	36
2.6 SOFTWARES ASSISTENTES VIRTUAIS DISPONÍVEIS NO MERCADO	39
2.7 CONSIDERAÇÕES FINAIS	42
<b>3 ARQUITETURA DO ASSISTENTE FINANCEIRO VIRTUAL</b>	<b>46</b>
3.1 PRESENTATION LAYER	48
<b>3.1.1 Speech to text</b>	<b>48</b>
<b>3.1.2 Text To Speech</b>	<b>48</b>
<b>3.1.3 TTS and STT Dictionaries</b>	<b>49</b>
<b>3.1.4 User Interface Manager</b>	<b>49</b>
3.2 ORCHESTRATION LAYER	50
<b>3.2.1 Coordinator</b>	<b>50</b>
<b>3.2.2 Dialog and Knowledge Manager</b>	<b>52</b>
3.3 UNDERSTANDING / KNOWLEDGE LAYER	53
<b>3.3.1 Natural Language Processor</b>	<b>54</b>
<b>3.3.2 Dictionary of The Financial Domain</b>	<b>54</b>
<b>3.3.3 Grammar of The Financial Domain</b>	<b>55</b>

<b>3.3.4 The Financial Information Services</b>	<b>57</b>
3.4 DATA LAYER	57
<b>3.4.1 XBRL Repositories</b>	<b>58</b>
3.5 FLUXO DE INFORMAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL	58
3.6 CONSIDERAÇÕES FINAIS	62
<b>4 IMPLEMENTAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL</b>	<b>63</b>
4.1 IMPLEMENTAÇÃO DA PRESENTATION LAYER	65
<b>4.1.1 Aplicação cliente móvel</b>	<b>65</b>
<b>4.1.2 Aplicação cliente em página HTML/Javascript</b>	<b>66</b>
<b>4.1.3 Tratamento multilíngue dos clientes</b>	<b>68</b>
4.2 IMPLEMENTAÇÃO DA ORCHESTRATION LAYER	69
<b>4.2.1 Classe Coordinator</b>	<b>71</b>
<b>4.2.2 Classe AssistantContext</b>	<b>72</b>
<b>4.2.3 Classe RequestParameters</b>	<b>74</b>
<b>4.2.4 Classes NaturalLanguageProcessor e NLPServiceAdapter</b>	<b>75</b>
<b>4.2.5 Classe AssistantResponse</b>	<b>79</b>
<b>4.2.6 Classe DialogKnowledgeManager</b>	<b>79</b>
<b>4.2.7 Classe Service Response</b>	<b>82</b>
4.3 IMPLEMENTAÇÃO DA UNDERSTANDING / KNOWLEDGE LAYER	83
<b>4.3.1 Natural Language Processor</b>	<b>83</b>
<b>4.3.2 Implementação do dicionário de termos</b>	<b>87</b>
<b>4.3.3 Regras gramaticais</b>	<b>94</b>
<b>4.3.4 Reconhecimento das intenções do usuário</b>	<b>95</b>
<b>4.3.5 Implementação do Financial Information Services</b>	<b>100</b>
<b>4.3.6 Classe FinancialInformationService</b>	<b>104</b>
4.4 IMPLEMENTAÇÃO DA CAMADA DATA	109
<b>4.4.1 Classe XbrlUsgaapWebService</b>	<b>109</b>
4.5 FUNCIONAMENTO INTEGRADO DOS COMPONENTES E SERVIÇOS	111
4.6 AMBIENTE DE DESENVOLVIMENTO DO ASSISTENTE	115
<b>4.6.1 Softwares utilizados</b>	<b>115</b>
<b>4.6.2 Hardwares utilizados</b>	<b>115</b>
4.7 CONSIDERAÇÕES FINAIS	116

<b>5 TESTES E AVALIAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL</b>	<b>118</b>
5.1.1 Testes de performance do Assistente Financeiro Virtual	118
5.1.2 Resultados dos testes de performance	119
5.1.3 Teste comparativo com os assistentes virtuais de mercado	121
5.1.4 Resultado do teste comparativo	121
5.2 CONSIDERAÇÕES FINAIS	124
<b>6 CONCLUSÃO</b>	<b>126</b>
6.1 PRINCIPAIS CONTRIBUIÇÕES	130
6.2 PRINCIPAIS RESTRIÇÕES E LIMITAÇÕES	131
6.3 PUBLICAÇÃO	132
6.4 TRABALHOS FUTUROS	133
<b>7 REFERÊNCIAS</b>	<b>134</b>
<b>APÊNDICE A - LISTA DE PERGUNTAS UTILIZADAS NOS TESTES</b>	<b>139</b>
<b>APÊNDICE B – AMOSTRA DE UMA MEDIÇÃO REALIZADA NO NAVEGADOR WEB</b>	<b>141</b>
<b>APÊNDICE C – PLANINHA DE DADOS COM RESULTADOS DAS MEDIÇÕES REALIZADAS COM O ASSISTENTE FINANCEIRO VIRTUAL</b>	<b>144</b>
<b>ANEXO A – XML DE RESPOSTA DE UM SERVIÇO DE DADOS EM XBRL</b>	<b>145</b>

## 1 INTRODUÇÃO

Este capítulo contextualiza o tema desta dissertação e apresenta as justificativas e a motivação para a sua elaboração. Além disso os objetivos da dissertação são detalhados e a metodologia utilizada também é apresentada.

### 1.1 CONTEXTUALIZAÇÃO

Em mundo globalizado como o de hoje a interconexão eletrônica, por meio das redes de telecomunicações e sistemas de informação, tem um papel importante, gerando impactos diversos na cultura dos povos, no conhecimento global, nas relações internacionais, na economia dos países. As variações no comportamento da economia de um país podem influenciar velozmente o comportamento da economia de outros países. Pequenas dificuldades financeiras em uma grande potência econômica reverberam rapidamente no sistema financeiro das nações parceiras. Os investidores atentos a esta realidade procuram conhecer diversos e distintos mercados e seus movimentos para então investir seu capital em empresas oriundas das mais distintas partes do globo. Eles buscam diversificar suas aplicações financeiras para aumentar seu patrimônio e/ou diminuir o risco de perdas provocadas pelos mais diversos motivos, como distúrbios em uma economia regional ou instabilidades políticas. Um mercado bastante utilizado por esses investidores é o mercado de ações, onde, hoje em dia, é possível atuar (operar) em quase todas as bolsas de valores espalhadas pelo mundo a partir de um sistema computacional com acesso a Internet. É um ambiente de alto risco financeiro, que pode trazer uma alta taxa de retorno sobre o investimento, assim como um grande prejuízo.

Para aumentar a taxa de sucesso nos investimentos nesse mercado é imprescindível que as análises e consequentes decisões sejam tomadas baseadas em informações corretas e mais atuais possíveis. Isto porque, em um determinado momento, as ações de uma empresa, podem estar com um preço menor do que de costume, indicando um bom momento de compra, entretanto, ao mesmo tempo, a saúde financeira dessa mesma empresa pode estar ruim. Neste cenário, se o investidor adquirir ações dessa empresa sem identificar a real situação econômica

dela, ele não estará sendo prudente pois estará ampliando o risco de perda de parte, o de todo, o seu capital investido na transação financeira.

Para auxiliar a tomada de decisão dos investidores em suas transações muitos deles consultam as mais diversas informações financeiras e contábeis sobre as empresas em que pretendem aplicar seu capital. A maioria dessas informações são divulgadas em relatórios na Internet pelas próprias empresas e grande parte deles em formato eletrônico Portable Document Format (PDF), texto, ou em formato HyperText Markup Language (HTML), adequados para serem lidos por pessoas. Quanto mais informações puderem ser consolidadas e processadas em um período de tempo inferior a validade delas, melhor será análise para tomada de decisão, tarefa que é delegada a uma unidade computacional. Entretanto, os formatos citados anteriormente restringem a capacidade de processamento das informações por meio de computadores. Para que os dados apresentados sejam analisados por softwares, se faz necessária uma redigitação ou importação dos dados, embutindo o risco de ocorrência de falhas ou fraudes no processo de análise.

As dificuldades na troca de informações financeiras eletrônicas entre organizações foram bastante minimizadas com a criação da eXtensible Business Reporting Language (XBRL) (XBRL INTERNATIONAL, 2016b), um padrão aberto que engloba muitas tecnologias e permite representar um dado e o seu significado semântico no domínio financeiro. Com ela é possível a geração de documentos que são, ao mesmo tempo, relatórios com informações claras para os especialistas financeiros e uma fonte de dados estruturados e processáveis por computadores. Algumas empresas de consultoria de investimentos estão utilizando informações das corporações disponibilizadas em XBRL, motivadas principalmente pelas vantagens relacionadas a maior velocidade de troca de dados e a maior facilidade de detectar inconsistências nos relatórios.

Os serviços de assessoria financeira normalmente englobam levantamentos e análises personalizadas sobre uma grande quantidade de informações em um determinado espaço de tempo. Normalmente o produto final desses serviços é um relatório customizado de acordo com a solicitação dos clientes, e, geralmente, custa caro. De um modo geral, relatórios financeiros construídos de acordo com os critérios definidos por um cliente específico são solicitados por pessoas com grande capital para investir e que não possuem tempo disponível, ou interesse, para identificar e analisar os dados em demonstrativos financeiros disponíveis na Internet.

Para os pequenos investidores, sem capital suficiente para contratar uma assessoria e, também, sem tempo para ele mesmo coletar os dados e analisar as informações financeiras, o risco de se investir em empresas deficitárias e, conseqüentemente, obter prejuízo é bem maior.

Para conhecer a saúde econômica de uma organização em que se pretende realizar um investimento, alguns questionamentos sobre ela podem ser feitos. Questões do tipo: qual o nível de alavancagem de determinada empresa? Qual a evolução do endividamento da instituição Y nos últimos N anos? Qual o lucro líquido da empresa Z no segundo trimestre do ano X? Essas perguntas podem ser submetidas a um assistente, ou consultor financeiro, um profissional especialista em finanças corporativas que são remunerados pelos serviços prestados por demanda ou por disponibilidade.

## 1.2 JUSTIFICATIVA

Uma opção econômica para os investidores é obter informações financeiras de empresas por meio da utilização de um assistente virtual. Os smartphones disponíveis atualmente, em diversos sistemas operacionais, oferecem esse serviço, e.g. Siri (APPLE, 2016d), Cortana (MICROSOFT, 2016b), Google Now (GOOGLE, 2016e). Esses assistentes são sistemas computacionais proprietários que respondem as perguntas dos usuários interagindo com eles por meio de linguagem natural, e até por interação por voz, de forma mais fácil de ser realizada do que através das tradicionais interações com os sites web. Para Zambiasi (2012a, p. 250), na busca por uma interface mais amigável com o usuário, “é mais intuitivo e natural o usuário “conversar” com seu assistente sem a necessidade de teclado, mouse ou monitor”. Entretanto, até o presente momento, as respostas disponibilizadas por esses assistentes não atendem satisfatoriamente aos questionamentos financeiros, pois são respondidos com informações incompletas e superficiais, normalmente apenas com indicações de sites especializados sobre o assunto. O Siri eventualmente fornece respostas financeiras muito próximas do que é perguntado. Uma das principais razões para esses assistentes não fornecerem as respostas diretas e com mais precisão sobre a situação financeiras das empresas se deve ao fato deles serem assistentes de propósito geral. Isto é, são assistentes construídos para fornecerem informações ou executar ações úteis, mas de contextos do

cotidiano, e.g. informar as condições do clima, informar como chegar a determinada localização, enviar uma mensagem, agendar um compromisso.

Supondo que as informações fornecidas por esses assistentes fossem coerentes e completas para o ambiente financeiro, surge outra questão que está relacionada à sua credibilidade. Atualmente, as empresas proprietárias dos assistentes mais utilizados, também vendem informações relacionadas ao perfil de uso dos seus clientes para seus parceiros comerciais, o que evidencia o alto risco de influência dos patrocinadores sobre os dados retornados.

Um sistema computacional especializado no domínio financeiro e que presta serviço de assistência financeira, através de respostas a questionamentos especializados utilizando linguagem natural, pode ser uma alternativa efetiva à assessoria financeira do tipo pergunta e resposta para o investidor. Além disso, esse sistema deve utilizar, como base para as suas respostas, dados corporativos disponibilizados por instituições de controle financeiro reconhecidas internacionalmente pela sua confiabilidade e que possibilitem que sistemas automatizados identifiquem fraudes ou inconsistências nos mesmos. Portanto, os atuais assistentes virtuais não se caracterizam como essa alternativa.

Portanto, definiu-se os seguintes requisitos para balizar a proposta de desenvolvimento do Assistente Financeiro Virtual:

- a) responder questões elaboradas pelo usuário relacionadas a situação financeira de empresas de forma direta e precisa.
- b) interagir com usuário através de linguagem natural reconhecendo e respondendo aos seus questionamentos feitos utilizando o jargão financeiro, a fim de facilitar para o usuário, que não deverá necessitar de nenhum conhecimento técnico prévio especializado na utilização do assistente.
- c) manter um diálogo de perguntas e respostas com o usuário, onde as respostas devem ser dadas em um tempo inferior a cinco segundos e as perguntas subsequentes podem ocultar informações já fornecidas pela pergunta anterior feita pelo usuário. O objetivo é simular uma conversa entre pessoas (cliente e consultor) e diminuir o esforço do usuário em realizar perguntas completas (com todas as informações) a cada interação.

- d) utilizar como fonte da informação, dados públicos em formato padronizado disponibilizados por uma instituição de controle ou órgão financeiro reconhecido internacionalmente.
- e) ser de propósito específico, e domínio controlado, no caso, o domínio financeiro.
- f) utilizar a tecnologia web, pois esta está disponível em uma ampla quantidade de dispositivos e plataformas de software.

### 1.3 MOTIVAÇÃO

Após a coleta de informações preliminar em busca do estado da arte nas áreas de assistentes virtuais, processamento de linguagem natural e manipulação e apresentação de dados financeiros, percebeu-se que não foi encontrado nenhum estudo relacionado as três áreas de forma concomitante. Isto é, não foi encontrada nenhuma evidência no meio científico, e nem na iniciativa privada a respeito de assistentes virtuais financeiros.

Na área da manipulação de dados financeiros foram encontrados diversos trabalhos que abordavam a tecnologia XBRL. Já para a área de processamento de linguagem natural muitos avanços foram obtidos nos últimos anos, principalmente com o uso de modelo por serviços. Para a área dos assistentes virtuais, os progressos também aconteceram, principalmente na subárea de assistentes virtuais de propósito geral.

O desconhecimento da existência de um trabalho científico relacionado a um assistente virtual financeiro motivou a elaboração desse trabalho. Após o levantamento bibliográfico, e diante de todos os requisitos do assistente citados anteriormente, selecionou-se as seguintes tecnologias para implementação:

- a)XBRL, que disponibiliza dados financeiros que geram formulários inteligíveis por pessoas e, de forma concomitante, podem ser processáveis por computadores. Ela permite a utilização de taxonomias para troca padronizada de informações entre instituições, e.g. U.S. Generally Accepted Accounting Principles (US-GAAP) (XBRL US, 2016a), que é uma taxonomia que corresponde aos princípios contábeis geralmente aceito no mercado financeiro americano ou a International Financial Report Statement (IFRS) (AICPA, 2016), que é uma taxonomia que está sendo

adotada como padrão mundial para normas Internacionais de contabilidade;

- b) Processamento de linguagem natural para domínio customizado, que permite a construção de um vocabulário específico do mercado financeiro para reconhecimento das perguntas dos usuários do assistente financeiro virtual;
- c) Arquitetura orientada a serviços que proporciona, através do baixo acoplamento, a interação de distintas plataformas de software.

Este trabalho busca responder as seguintes questões de pesquisa:

- 1.Os assistentes virtuais disponíveis atualmente podem ser customizados para responder questões de um domínio específico?
- 2.Os frameworks ou arquiteturas disponíveis permitem gerar assistentes virtuais de domínio restrito do tipo pergunta e resposta?
- 3.O Assistente Financeiro Virtual pode ser totalmente baseado em uma arquitetura orientada a serviço, inclusive utilizando um serviço de processamento de linguagem natural?
- 4.É possível realizar consultas de informações financeiras em dados disponibilizados em XBRL utilizando linguagem natural? Em caso positivo, essa mesma consulta de dados disponibilizados em XBRL pode ser realizada em idiomas diferentes?
- 5.O Assistente Financeiro Virtual pode realizar análises financeiras complexas utilizando as informações registradas em relatórios em XBRL.
- 6.O Assistente Financeiro Virtual pode utilizar taxonomias financeiras diferentes?
- 7.Os serviços de processamento de linguagem natural disponíveis atualmente possuem tempos de resposta que permitem a construção de assistentes virtuais com tempos de resposta global compatíveis com os assistentes virtuais disponíveis no mercado?

#### 1.4 OBJETIVO

Objetivo geral desta dissertação é o desenvolvimento de um assistente virtual de propósito específico para o domínio financeiro que responde ao usuário sobre

questões financeiras relacionadas aos indicadores financeiros das empresas utilizando linguagem natural.

Para se alcançar este objetivo, alguns objetivos específicos são almejados:

1. Identificar frameworks ou arquiteturas para criação de assistentes virtuais para facilitar o desenvolvimento do estudo
2. Descrever a arquitetura utilizada para implementar o Assistente Financeiro Virtual, a fim de expor a solução tecnológica criada;
3. Implementar os artefatos de software e descrever a tecnologia utilizada, para entendimento de como foi realizada a implementação e como as tecnologias foram integradas;
4. Realizar estudo de caso e analisar os resultados da execução do Assistente, e.g. tempos de resposta e correção do Assistente para avaliar a eficácia da solução;
5. Comparar os resultados fornecidos com os resultados de outros assistentes disponíveis, a fim de permitir a avaliação da viabilidade do Assistente.

## 1.5 METODOLOGIA

Para facilitar a realização deste trabalho de pesquisa foi selecionada uma metodologia científica. A metodologia “é o estudo do método, ou seja, é o corpo de regras e procedimentos estabelecidos para realizar uma pesquisa” (TARTUCE, 2006 apud GERHARDT ; SILVEIRA, 2009, p. 11). Através da metodologia procurou-se identificar qual o método mais adequado a ser utilizado. Método é “o caminho em direção a um objetivo” (TARTUCE, 2006 apud GERHARDT ; SILVEIRA, 2009, p. 11). Metodologia Científica é “o estudo sistemático e lógico dos métodos empregados nas ciências, seus fundamentos, sua validade e sua relação com as teorias científicas” (TARTUCE, 2006 apud GERHARDT ; SILVEIRA, 2009, p. 11).

Para este trabalho foi utilizado o método hipotético dedutivo, que é adequado para explicar problemas onde o conhecimento disponível sobre o tema é insuficiente para explicar um fenômeno. Segundo Gerhardt e Silveira, (2009), para explicar os problemas, o método instrui a formular hipóteses das quais são deduzidas as consequências. Estas consequências devem ser testadas ou falseadas, isto é, deve-se procurar evidências empíricas que tornem falsas as hipóteses formuladas. Quando as hipóteses não conseguem ser falseadas, obtém-se a confirmação da

explicação do problema. Esta confirmação não é definitiva “pois a qualquer momento poderá surgir um fato que a invalide” (GERHARDT ; SILVEIRA, 2009, p. 27).

Como este trabalho busca a objetividade, a abordagem escolhida para pesquisa foi a quantitativa. Conforme Fonseca (2002, apud GERHARDT ; SILVEIRA, 2009, p. 33), na abordagem quantitativa, a compreensão do fenômeno é obtida através da análise de dados brutos coletados de forma neutra e padronizada. Este trabalho se configura como uma pesquisa aplicada, pois tem a pretensão de resolver um problema específico e para uma aplicação prática.

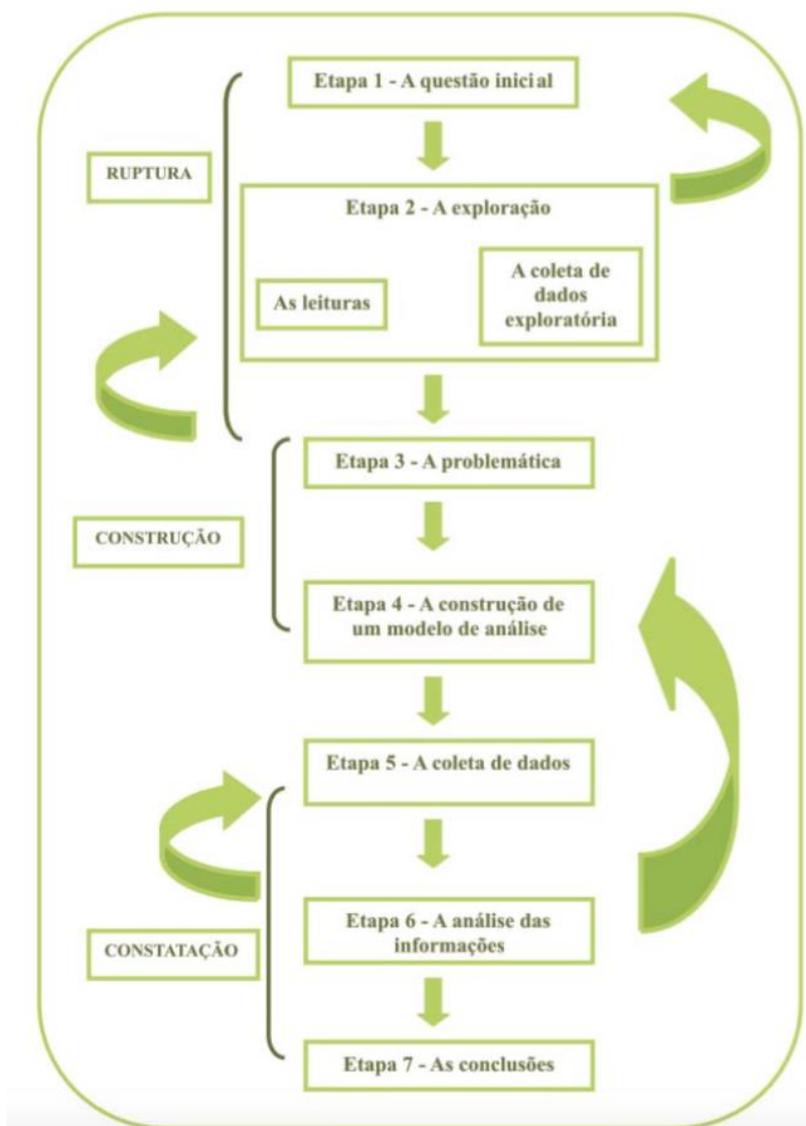
De acordo com os objetivos desta dissertação, ela é classificada como exploratória, pois busca explicitar o problema para construir as hipóteses. O procedimento científico escolhido para este trabalho foi o de levantamento, do tipo levantamento de uma amostra. A escolha deste procedimento foi razão das vantagens citadas por Gerhardt e Silveira (2009, p. 38), como “o conhecimento direto da realidade, economia e rapidez, e obtenção de dados agrupados em tabelas que possibilitam uma riqueza na análise estatística.”

Para facilitar a elaboração da pesquisa foi utilizado o esquema proposto por Quivy e Campenhoudt (1995 apud GERHARDT ; SILVEIRA, 2009), apresentado na Figura 1. As flechas de retroação na figura sinalizam a interação constante entre as etapas.

Com o objetivo de identificar os estado da arte na área de estudo, os seguintes sites de indexação de artigos científicos disponíveis na web foram consultados:

- ACM Digital Library- <http://www.acm.org/dl>
- Google Acadêmico- <http://scholar.google.com/>
- IEEE Xplore- <http://ieeexplore.ieee.org/>
- Mendeley - <https://www.mendeley.com/>
- Portal CAPES - <http://www.periodicos.capes.gov.br/>
- Science Direct - <http://www.sciencedirect.com/>
- Springer - <http://link.springer.com/>

Figura 1 - Etapas da pesquisa científica



Fonte: Gerhardt e Silveira (2009 apud QUIVY ; CAMPENHOUDT, 1995)

Na etapa de exploração, onde ocorre a coleta de dados e leitura de artigos científicos objetivando a definição do problema de pesquisa, foram construídas e utilizadas algumas *strings* de busca junto aos respectivos motores de busca dos sites de indexação de artigos científicos, conforme Quadro 1. Durante o processo de construção da problemática, as *strings* de busca utilizadas no levantamento bibliográfico sofreram algumas modificações, até a delimitação final do problema de pesquisa.

Na etapa de coleta exploratória de dados, os primeiros agrupamentos de artigos selecionados foram obtidos e administrados nas próprias bibliotecas digitais utilizando seus respectivos sites. As *strings* e bibliotecas utilizadas, bem como, a quantidade de artigos selecionados, estão apresentados no Quadro 1.

Quadro 1 - *Strings* de busca utilizadas nas respectivas bibliotecas digitais

String de busca	Biblioteca	Quantidade de artigos identificados	Quantidade de resumos de artigos lidos	Quantidade de resumos de artigos selecionados	Quantidade de artigos selecionados
((Natural and Language and Processing) and xbrl)	The ACM Guide to Computing Literature	59			
((Natural and Language and Processing) or "Virtual assistant") and xml	The ACM Guide to Computing				
("Virtual assistant")	The ACM Guide to Computing	177			
Abstract:"virtual assistant",	The ACM Guide to Computing	23	23	4	
ABSTRACT({virtual assistant})	Science Direct	3	3	2	
("Abstract":"virtual assistant"),	IEEE Xplore DL	2	2	2	2
"virtual assistant"	Mendeley	25	25	11	8
(Abstract:"personal assistant")	The ACM Guide to Computing Literature	122	122	20	9
("Abstract":"personal assistant")	IEEE Xplore DL	99	99	18 (7 em comum)	2
ABSTRACT({personal assistant})	Science Direct.	28	28	7 (2 em comum)	3
"personal assistant"	Springer	114	114	3	1
"personal assistant".	Google Scholar	25.000			
"virtual assistant" and XBRL	Google Scholar	2	2	0	0

String de busca	Biblioteca	Quantidade de artigos identificados	Quantidade de resumos de artigos lidos	Quantidade de resumos de artigos selecionados	Quantidade de artigos selecionados
"personal assistant" and XBRL	Google Scholar	4	4	1	1

Fonte: Próprio autor.

A soma total de artigos identificados pelas *strings* de busca correspondeu a 434. Este montante foi insumo para uma nova seleção de artigos que foi realizada através da leitura e avaliação dos respectivos resumos utilizando como critério de exclusão as referências dos artigos cujos resumos não possuíam nenhuma relação com a área ou tema da dissertação.

As referências selecionadas foram incluídas em um repositório único de documentos e foram classificadas por subtemas relacionados ao trabalho, como: assistência pessoal, assistência virtual, XBRL, processamento de linguagem natural. O objetivo em incluir as referências neste repositório foi facilitar a gestão das referências na dissertação. A ferramenta utilizada como repositório foi o Mendeley.

Das 434 referências iniciais, 21 foram selecionadas e incluídas no repositório por ter relação direta com o tema da dissertação. Das 21 referências selecionadas, não foi possível obter os artigos de 7, por não haver permissão de acesso na web. Foram obtidos 14 artigos, que foram lidos em sua plenitude e serviram de base para este trabalho.

Além da pesquisa bibliográfica, foram utilizados também os principais buscadores da web, e.g. Google (2016a), Bing (MICROSOFT, 2016a), Yahoo (2016), a fim de identificar trabalhos e artefatos de software implementados, seja *open source* ou proprietários, relacionados a assistentes virtuais e/ou pessoais.

## 1.6 Organização da Dissertação

Este trabalho está organizado em seis capítulos. Neste capítulo o tema da dissertação foi contextualizado. Dentro desse contexto, alguns conceitos foram preliminarmente apresentados, como o conceito de assistente virtual, bem como foram mostradas as funcionalidades que devem ser fornecidas por um assistente financeiro virtual. Além disso, a motivação para realização deste trabalho, as questões de pesquisa, os objetivos da dissertação e a metodologia de pesquisa

científica utilizada para o desenvolvimento deste trabalho foram, também, apresentados.

No capítulo dois são apresentados os conceitos relacionados ao termo Assistente Virtual e o resultado de uma revisão da literatura sobre trabalhos científicos e alguns produtos de software disponíveis no mercado relacionados, principalmente, à assistentes virtuais de propósito específico. São analisadas as contribuições e pontos de melhoria, dos trabalhos e produtos de software aderentes a criação de um Assistente Financeiro Virtual.

O terceiro capítulo apresenta a proposta de uma nova arquitetura para construção de assistentes virtuais de propósito específico, que é a arquitetura do Assistente Financeiro Virtual. As camadas da arquitetura e seus respectivos serviços e componentes, bem como, as funções e contribuições de cada um deles para execução do Assistente, são apresentados em detalhes. Para demonstrar como ocorre a integração dos serviços e componentes da arquitetura, um fluxo de informação é descrito.

No quarto capítulo são apresentadas as implementações das camadas do Assistente Financeiro Virtual, que são baseadas na arquitetura apresentada no capítulo três. A justificativa pela escolha da tecnologia utilizada em cada componente e serviço implementado e, os detalhes de como a implementação foi realizada, são mostrados nesse capítulo. A seleção do serviço de processamento de linguagem natural e a sua configuração, envolvendo dicionários e gramáticas associadas, para realização do reconhecimento das perguntas financeiras do usuário, também são apresentadas.

O capítulo cinco expõe como foram realizados os testes de correção e desempenho do Assistente Financeiro Virtual. Esse capítulo apresenta a avaliação dos resultados e um comparativo entre as respostas fornecidas pelo Assistente Financeiro Virtual e as fornecidas por um software assistente virtual disponível no mercado.

Na conclusão são apresentadas as considerações finais sobre o Assistente Financeiro Virtual e as respostas às questões de pesquisa. As principais contribuições e limitações deste trabalho, além de sugestões para trabalhos futuros também são apresentadas nesse capítulo.

Por fim, no apêndice A estão apresentadas as perguntas que foram submetidas, em sequência, à implementação do Assistente Virtual Financeiro para

realização dos testes e avaliação dos resultados. No apêndice B está apresentada parte de uma amostra de medição dos testes com o Assistente Financeiro Virtual. O apêndice C ilustra a planilha dos dados das medições realizadas nos testes. O anexo A mostra um arquivo XML que contém uma resposta de um dos serviços de mercado utilizados pelo Assistente Financeiro Virtual.

## 2 ASSISTENTES VIRTUAIS

Este capítulo tem como objetivo identificar os conceitos e apresentar e analisar os trabalhos científicos e implementações relacionados aos assistentes virtuais identificados no levantamento bibliográfico realizado.

Nesse levantamento percebeu-se que o termo Assistente Virtual (AV), ou em inglês, *Virtual Assistant*, pode ter diversas nomenclaturas e denominações, o que fez necessária a ampliação dos termos da pesquisa. Para Paraiso e Barthès (2005, p. 496), *Personal Assistants* são agentes que auxiliam pessoas a realizar seus trabalhos diários. Já em um estudo mais recente, Zambiasi e Rabelo (2012, p. 1227) denominam como Software Assistente Pessoal (SAP), em inglês *Personal Assistant Software* (PAS), os *softwares* que auxiliam pessoas nas suas atividades diárias em casa ou no trabalho. Eles afirmam que este tipo de *software* possui várias denominações e características identificadas por vários autores e dificilmente chega-se a uma única definição.

Para Zambiasi (2012, p. 98) um assistente pessoal é um conjunto de processos computacionais operando de forma colaborativa criado para representar o usuário na execução de determinadas tarefas de forma automática ou semi-automática.

Eisman, López e Castro (2012, p. 3136) utilizam o termo *Embodied Conversational Agent* (ECA), ou Agente de Conversação Embutido. Eles definem como um sistema inteligente representado por um personagem capaz de se envolver em uma conversação como uma pessoa. Este agente tem como principal função auxiliar o usuário a realizar determinadas tarefas.

As diversas denominações estão diretamente atreladas às ações realizadas pelos assistentes. Alguns requisitos definidos para considerar se o assistente é pessoal ou virtual, nos trabalhos publicados, chegam a ser conflitantes e excludentes, e. g. para Zambiasi (2012), todo assistente pessoal deve ser de propósito geral, não específico à apenas uma atividade, definição que difere do assistente pessoal apresentado por Paraiso e Barthès (2005), que foi projetado exclusivamente para orientar os usuários, de um site governamental, a como solicitar serviços. O propósito de auxiliar o usuário a realizar determinada tarefa é um consenso.

Medina, Eisman e Castro (2013) distinguem os assistentes como de propósito geral e de domínio específico. Os assistentes de propósito geral são úteis para fornecer informações sobre assuntos corriqueiros, e.g. obter informações sobre o clima, sobre como chegar a determinada localização, executar algumas ações como agendar um compromisso, elaborar e enviar uma mensagem. Os assistentes de propósito específico servem para responder questionamento a respeito de um domínio específico, e.g. uma loja, um museu, uma instituição, uma área do conhecimento.

Kavitha e Pande (2013, p. 1) afirmam que na área de Assistentes Pessoais muitos sistemas de diálogo (Dialogue System), que são sistemas de conversação que permitem que uma pessoa se comunique com um computador usando linguagem natural, foram desenvolvidos ao longo dos anos e em vários domínios do conhecimento. A maioria desses sistemas foi desenvolvida para facilitar a extração de informações do banco de dados embutido nos respectivos sistemas.

Para Zhang et al (2015), os sistemas de compreensão da linguagem falada (Spoken Language Understanding - SLU) agora estão amplamente disponíveis e comumente usados em uma variedade de áreas de aplicação, e.g., a área dos assistentes pessoais embutidos nos smartphones. Zhang et al (2015, p. 5306) afirma que esses assistentes são geralmente suportados por uma coleção de serviços on-line que executam tarefas computacionais tais como reconhecimento de voz e compreensão da linguagem natural e fornecem acesso a serviços de resposta à perguntas ou serviços de pesquisa na web.

Foi realizada uma revisão da literatura sobre determinados trabalhos considerados importantes e aderentes à proposta de construção de um Assistente Financeiro Virtual (AFV). A seguir são citados e analisados os trabalhos correlatos ao presente estudo.

## 2.1 FRAMEWORK PARA INTERAÇÃO POR VOZ

Martin et al (1996) apresentaram um framework, chamado SpeechActs, para facilitar a construção de aplicações controladas pelo usuário através de linguagem natural. Esse framework inclui um servidor de áudio, um processador de linguagem natural, um gerente de conversa (discourse manager), um conversor de texto em voz, além de um conjunto de ferramentas para apoiar a construção do domínio de conhecimento.

O framework contempla um dicionário que deve conter todas as palavras, e suas diversas formas, do domínio da aplicação. Esse dicionário pode conter a definição de sinônimos para cada palavra, a fim de ampliar o vocabulário que será reconhecido pela aplicação.

O framework define uma gramática de reconhecimento de voz, determina que palavras foram faladas pelo usuário, e uma gramática de processamento de linguagem natural que extrai os significados dessas palavras. Essas gramáticas apoiam, respectivamente, o reconhecedor de voz e o processador de linguagem natural.

O gerente de conversa embutido no framework mantém uma fila com informações de contexto e uma pilha (The discourse stack) com as informações da conversa atual entre o computador e o usuário, cujo objetivo é simular o fluxo natural de uma conversa. Os dados inclusos, na pilha e na fila, permitem a aplicação identificar referências embutidas em pronomes inseridas nas expressões do usuário, i.e., “Me envie um sinal de alerta”, bem como, identificar referências implícitas em informações parciais proferidas pelo usuário, i.e., “e outro daqui a duas horas”.

## 2.2 FRAMEWORK PARA ASSISTENTE MÓVEL SENSÍVEL AO CONTEXTO

Chihani et al. (2013) apresentaram um framework para construir um assistente virtual para dispositivos móveis que, baseado em informações de contexto, auxilia o usuário na organização das atividades diárias. Para realizar as tarefas, esse framework segue regras que determinam as combinações entre aplicações instaladas no dispositivo móvel e quando cada combinação de aplicações deve ser invocada.

Para Chihani et al. (2013, p. 111) a principal limitação dos sistemas baseados em regras é que as linguagens para construção das regras disponíveis, são em sua maioria, muito complexas para usuários com conhecimentos básicos em programação de computadores. Partindo desse pressuposto, eles projetaram o framework em que a construção das regras é feita em linguagem natural.

Para que o assistente auxilie o usuário a realizar suas atividades diárias, i.e., enviar uma mensagem para uma pessoa quando determinada situação acontece, o usuário deve definir uma regra que define a atividade, que corresponde a informar quais, e como as, aplicações do dispositivo móvel serão acionadas, e quando essa

atividade será invocada. Assim, através da utilização de linguagem natural, o usuário define o comportamento do assistente em resposta às informações de contexto.

### 2.3 ASSISTENTE VIRTUAL COM INTERAÇÃO POR VOZ BASEADO EM AGENTES

Paraiso e Barthès (2005) descrevem o projeto de um assistente pessoal com interação por voz que auxilia usuário em um sistema governamental. Uma das principais preocupações foi a interação facilitada entre homem e máquina, permitindo o uso confortável e fácil para usuários menos experientes com computadores através de diálogos em linguagem natural. Os autores criaram um Assistente Pessoal que é utilizado como interface entre os usuários e um sistema multi-agente.

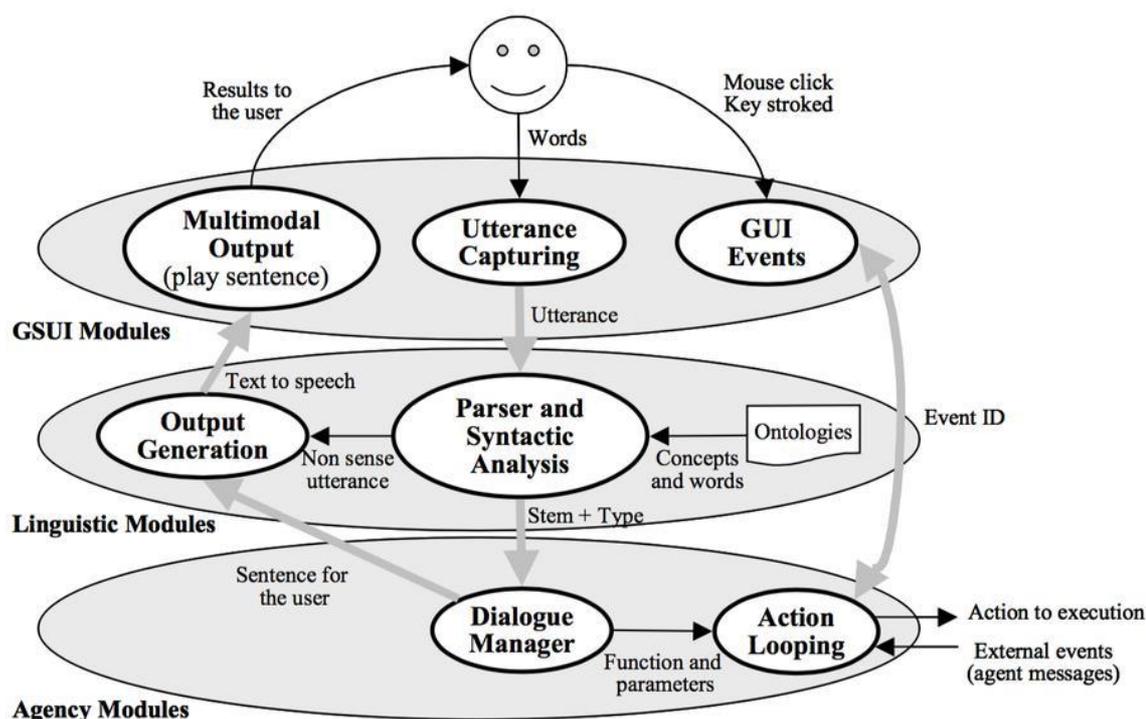
A arquitetura proposta por Paraiso e Barthès (2005) é dividida em três módulos, conforme Figura 2, que possuem as seguintes funções :

- interação com o usuário, realizada através de módulos *Graphic Speech User Interface* (GSUI) que coletam dados do usuário e produzem saídas multimodais;
- processamento das expressões, realizada pelos *Linguistic Modules* (LU) que interpretam as interações do usuário por meio de análise sintática e conversão apoiada por ontologias;
- gestão de diálogo, realizada pelos *Agency Modules* (AM) que aciona a execução de eventos externos por meio dos agentes e controlam o diálogo e a assistência.

O correto processamento das declarações do usuário e a coerente tomada de decisão dependem do conhecimento de domínio, isto é, a manipulação desse conhecimento é crucial para a eficácia da arquitetura proposta, responsabilidade atribuída às ontologias que permitem o compartilhamento de conhecimento e a reutilização.

As ontologias foram utilizadas para auxiliar a interpretação do contexto das mensagens enviadas por outros agentes e para manter a representação computacional do conhecimento útil no momento da inferência. Eles separaram os modelos de domínio e de tarefa, diminuindo a complexidade dos *Linguistic Modules*.

Figura 2 - Arquitetura de Assistente pessoal com interação por voz baseado em agentes

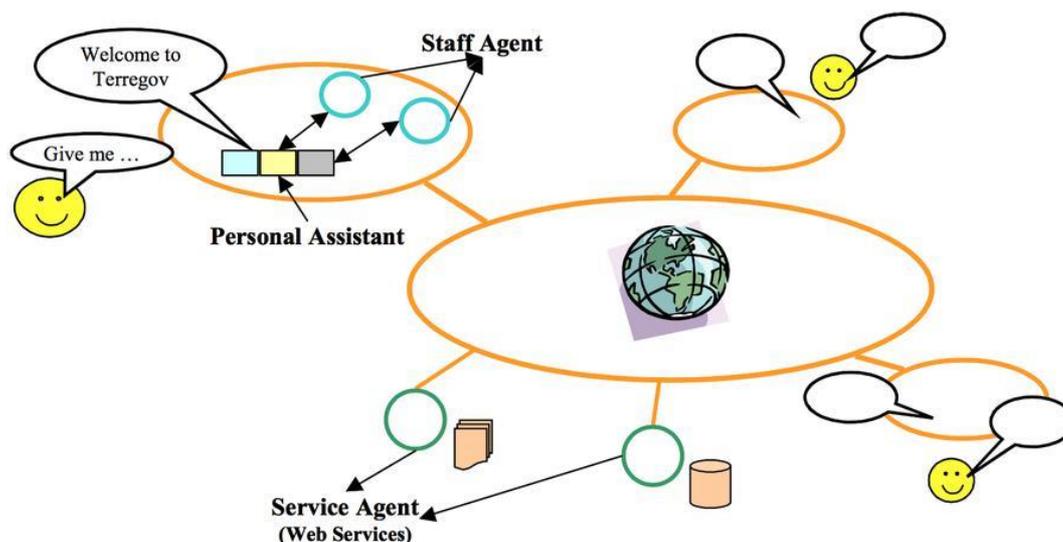


Fonte: Paraiso e Barthès (2005).

Os próprios autores desenvolveram a interface de conversação vocal que permite que os usuários interajam com o software utilizando seus próprios termos além de uma taxonomia para sustentar os enunciados dos usuários, permitindo analisar se a sentença está bem formada e de acordo com a estrutura gramatical, ou se está fora do domínio.

Para cada usuário é destinado um assistente pessoal que pode acionar agentes de serviço localmente. Estes, por sua vez, podem delegar sub-tarefas para outros agentes de serviços para completar uma tarefa complexa. Os agentes foram implementados e encapsulados por meio de *web services*. É importante destacar que os agentes são independentes e trocam informações entre eles e com o assistente pessoal. Os usuários nunca trocam informações com os agentes, somente com o assistente a que lhe foi destinado. Esse processo é mostrado na Figura 3.

Figura 3 - Processo de comunicação no assistente multiagente



Fonte: Paraiso e Barthès (2005).

## 2.4 FRAMEWORK PARA CONSTRUÇÃO DE ASSISTENTES VIRTUAIS DE DOMÍNIO RESTRITO

Eisman, López e Castro (2012), apresentam um *framework* para projetar assistentes virtuais de domínio fechado multilíngue que podem ser integrados a sites web já existentes. Uma das principais motivações do trabalho deles é oferecer uma alternativa a tradicional interação dos usuários com sites web, geralmente realizada por meio de cliques de mouse em menus estáticos e por buscas com a utilização de palavras chaves. Os assistentes criados por este *framework* têm como função principal auxiliar os usuários a encontrar informações úteis sobre determinados tópicos em um site web. A característica principal dele é a prestação de assistência à navegação em sites web à determinados domínios.

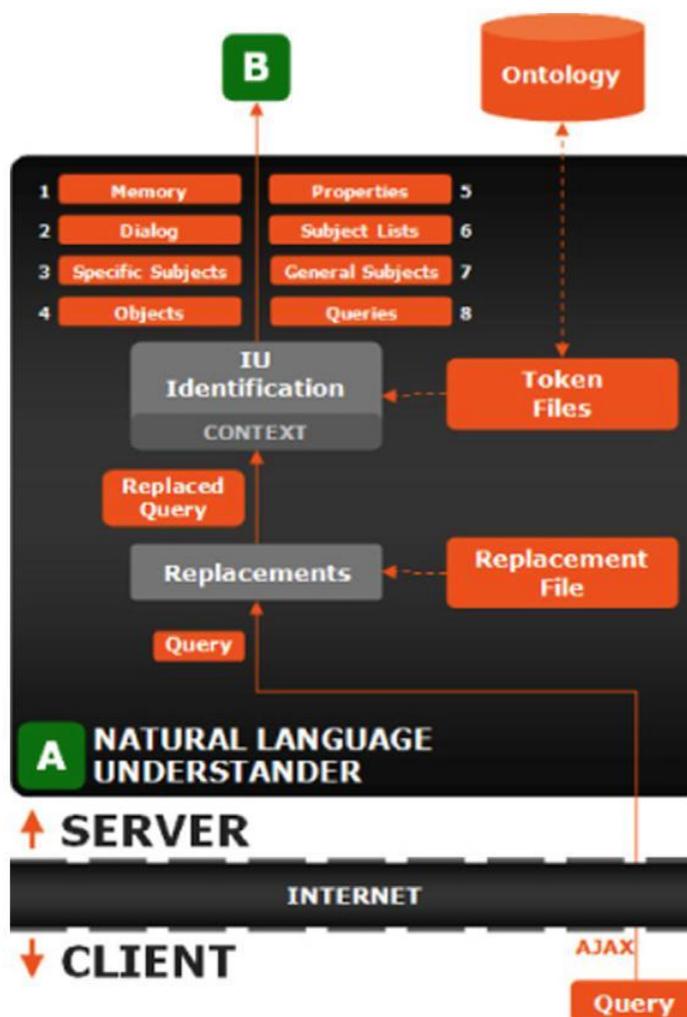
Foi apresentada a implementação de assistente virtual, baseado no *framework*, que fornece informações sobre uma determinada universidade respondendo a questionamentos que foram feitos pelo usuário em linguagem natural.

O conhecimento, ou o domínio, é armazenado em expressões regulares e em uma ontologia. As expressões regulares têm a função de facilitar o reconhecimento das questões feitas pelo usuário. Já a ontologia dá suporte ao processo de escolha da próxima ação a ser realizada e à geração da resposta adequada.

Segundo os autores a ontologia é uma representação formal de um grupo de conceitos dentro de um domínio particular e de alguns relacionamentos entre estes conceitos. A ontologia que eles criaram é baseada em seis diferentes entidades

chamadas Information Units (IU), conforme ilustrado na Figura 4, que correspondem a informações sobre um conceito específico. Além de conter a definição do conceito, essas entidades contêm meta-informações complementares que não estão isoladas e podem ser conectadas umas com as outras para construir relacionamentos.

Figura 4 - Processo de identificação das Information Units



Fonte: Eisman et al (2012).

As entidades criadas para atender o domínio de uma determinada universidade foram: Objetos, Propriedades, Assuntos específicos, Assuntos gerais, Listas de assuntos, Visitas guiadas.

Além do uso de ontologia, outra manipulação do conhecimento é feita por meio da utilização de expressões regulares. Elas são armazenadas em arquivos texto e contêm *tokens* associados as IUs, que facilitam o reconhecimento das diferentes questões feitas pelos usuários. As expressões regulares permitem presumir possíveis erros ortográficos, sinônimos ou, por exemplo, nomes UI

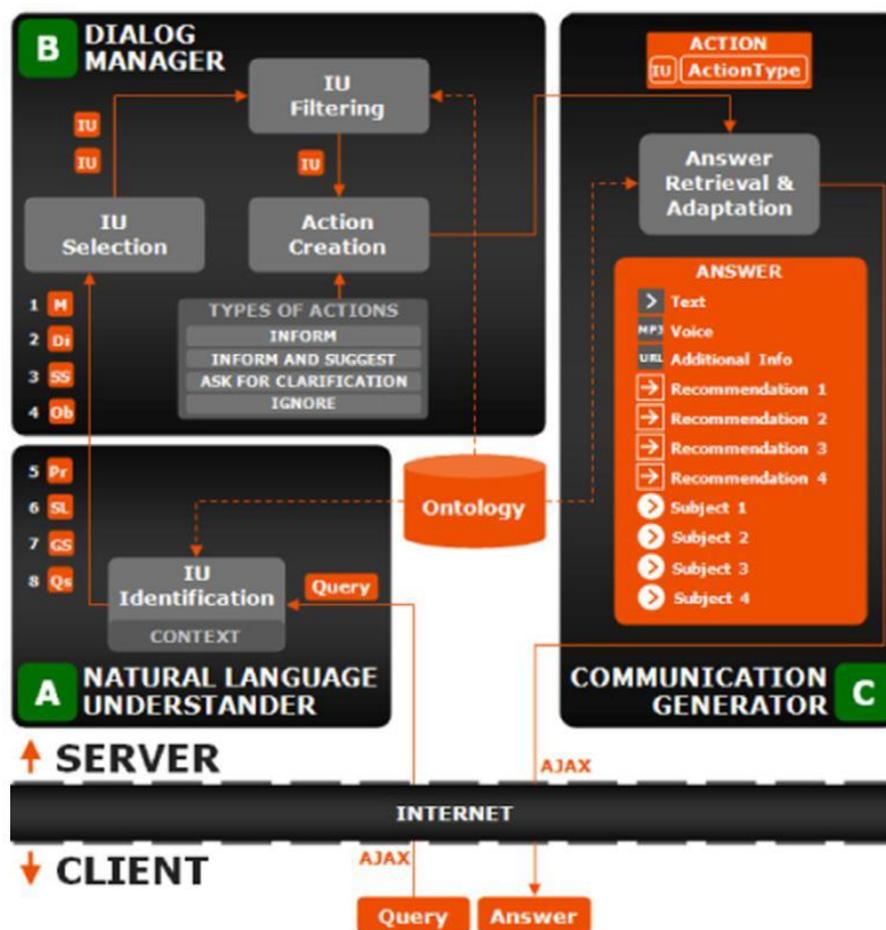
complexos, que podem aparecer na sentença elaborada pelo usuário. Este processo reduz a ambiguidade das questões.

O assistente proposto por Eisman, López e Castro (2012) foi projetado utilizando uma arquitetura cliente-servidor composta pelos seguintes módulos, conforme Figura 5:

- The Natural Language Understander (NLU), módulo responsável por reconhecer as questões elaboradas pelo usuário. A entrada de dados nesse módulo é uma questão elaborada pelo usuário em linguagem natural e a saída é uma lista de todas as Information Units reconhecidas na questão;
- The Dialog Manager (DM), módulo que determina o que, quando e como o assistente deve fazer internamente para responder a interação do usuário. A sua entrada é a lista de Information Units fornecidas pelo Natural Language Understander. Baseado em um grupo de regras, o DM analisa e seleciona quais IUs serão utilizados para gerar a resposta. Como saída, ele escolhe qual o tipo de ação abstrata será executada;
- The Communication Generator (CG), módulo responsável por gerar uma resposta específica que abrange a ação. A entrada é uma ação abstrata fornecida pelo Dialog Manager e a saída é uma resposta específica. O núcleo da resposta é um pedaço de texto associado à uma Information Unit que foi selecionado pelo DM.

O *framework* proposto por eles permite a construção de assistentes multilíngues que geram respostas textuais (que podem ser transformadas em voz), páginas web (onde o usuário pode encontrar mais detalhes), ou listas de recomendações ou de assuntos.

Figura 5 - Arquitetura do Framework baseado em Natural Language Understander



Fonte: Eisman et al (2012).

Os autores apresentaram a implementação de um assistente virtual, chamado Elvira, que foi implementado baseado no *framework*. Ele é representado por uma personagem que esboça expressões faciais e fornece informações sobre os cursos e serviços oferecidos pela Universidade de Granada na Espanha. O assistente está em funcionamento desde de fevereiro de 2010. As respostas das perguntas são respondidas em linguagem natural seguidas por um menu de links relacionados ao assunto perguntado. No estudo eles apresentam os resultados favoráveis da utilização da assistente no site web da citada Universidade.

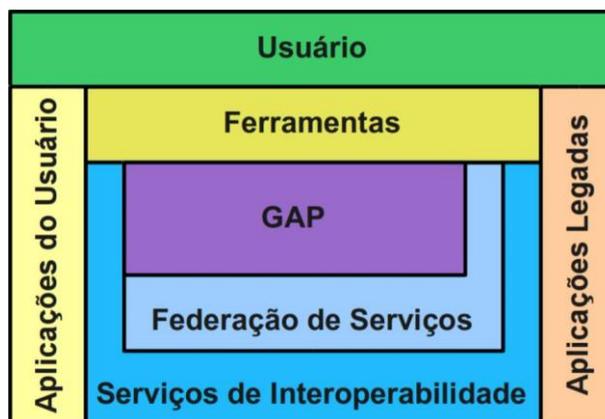
## 2.5 SOFTWARE DE ASSISTÊNCIA PESSOAL ORIENTADO A SERVIÇO

Zambiasi e Rabelo (2012) apresentaram um modelo conceitual e uma arquitetura de referência para Softwares Assistentes Pessoais (SAP). Os requisitos desejáveis de um SAP almejado pelos autores definem que o software deve ser interagente e integrado a ambientes de negócios corporativos, adaptativo, e interoperável com vários subsistemas e fontes de informação e de atividades.

Definem também que o SAP deve utilizar padrões abertos e flexíveis da indústria. O modelo e arquitetura apresentados são baseados no paradigma da arquitetura orientada a serviços – Service Oriented Architecture (SOA).

Zambiasi (2012) define que o SAP é um entidade em execução autônoma para auxílio a tarefas diárias do usuário. Em sua tese de doutorado ele apresenta uma alteração na arquitetura de referência, mas sem prejuízo ao suporte dos requisitos definidos para um SAP, conforme definido no trabalho discutido na Seção 2.4. A arquitetura proposta, conforme Figura 6, permite que o usuário interaja com o assistente por meio de Ferramentas, Aplicações Legadas ou Aplicações do Usuário. O elemento principal, o Gerenciador de Assistentes Pessoais (GAP) só utiliza serviços disponibilizados pela Federação de Serviços e para ocorrer a interação indireta com o usuário ou outros serviços incompatíveis existe a camada de Serviços de Interoperabilidade. Ambientes de desenvolvimento e ferramentas para desenvolver e configurar elementos da arquitetura compõe a camada Ferramentas.

Figura 6 - Arquitetura Orientada a Serviços do SAP



Fonte: Zambiasi (2012).

Um comportamento é uma composição selecionada de funcionalidades distribuídas em um conjunto maior disponível na internet. Estas funcionalidades, representadas pelos serviços, podem ser executadas em paralelo. A manipulação do conhecimento é representada pelos comportamentos embutidos do SAP. O comportamento do SAP é uma “composição de serviços disponíveis em repositórios distribuídos na Internet e que são escolhidos pelo usuário para compor as atividades que o assistente pode efetuar” (ZAMBIASI, 2012, p. 102).

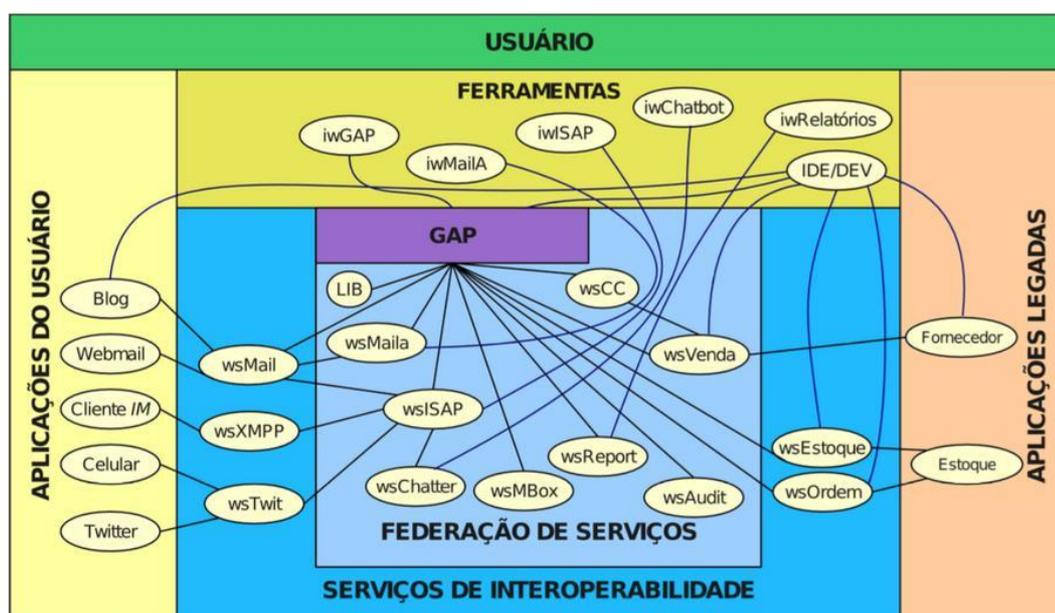
A arquitetura proposta facilita a adição de novos comportamentos e execução de atividades ao SAP de forma dinâmica. Ela permite, por exemplo, que os

comportamentos sejam criados e disponibilizados por distintos provedores de forma distribuída, abrindo a possibilidade de eles serem alugados ou até vendidos.

O autor apresentou uma implementação para verificar a eficácia da arquitetura proposta. O SAP instanciado teve a função de ajudar um funcionário a gerenciar um estoque e a criar um determinado relatório. Na Figura 7 é possível visualizar a composição de serviços, as aplicações de usuário, as aplicações legadas e a coordenação realizada pelo GAP.

A assistência que será realizada pelo SAP proposto vai depender dos comportamentos que ele possui, e estes, por sua vez, são construídos através da orquestração dos serviços selecionados disponíveis. Não há ontologia de domínio que sirva como base para manipulação do conhecimento e definição das ações de assistência.

Figura 7 - Composição dos elementos da implementação da arquitetura de referência



Fonte: Zambiasi (2012).

O SAP especificado objetiva atuar principalmente em um ambiente corporativo e se adequar aos diversos processos e transações empresariais. A possibilidade do usuário poder ter um assistente próprio, com os mais variados tipos de comportamentos, gostos e custos é uma característica bastante relevante.

## 2.6 SOFTWARES ASSISTENTES VIRTUAIS DISPONÍVEIS NO MERCADO

Na área de estudo de software de assistência virtual não somente há avanços importantes na comunidade acadêmica como também nas empresas privadas de desenvolvimento de software. Atualmente é possível perceber a utilização de assistentes virtuais nas mais diversas situações, como em sites de redes sociais, sites de compra de carros, livros e embutidos nos sistemas operacionais dos smartphones. Nesta seção, serão apresentados e discutidos alguns destes assistentes, indicando quando pertinente, as contribuições que eventualmente possam oferecer a esta dissertação.

O Sys Virtual Assistant (2015c), da empresa Synthetic Intelligence Network (2015b), é um software assistente virtual de propósito geral, cujo principal objetivo é auxiliar os usuários na interação homem-máquina, *human-computer interaction* - HCI. Este auxílio inclui tarefas como gerenciar lembretes, obter notícias atualizadas, verificar e enviar mensagens eletrônicas, apresentar relatórios sobre o clima, agendar reuniões.

Este assistente virtual disponibiliza avatares (figuras gráficas) 3D com expressões faciais que imitam personagens humanos. Segundo a empresa, esse assistente prima pela privacidade e segurança dos dados do usuário pois funciona completamente na máquina local, sem enviar dados para nenhum servidor na rede, característica que ele considera incomum entre os assistentes disponíveis no mercado.

A arquitetura desse assistente é baseada em um *Natural Language Processor* (NLP) que interage com uma coleção de módulos, ou *plugins*, sendo possível adicionar outros *plugins* responsáveis por novas funcionalidades. Esta possibilidade de modificação, inclusão ou exclusão de módulos permitem a um desenvolvedor configurar o comportamento do Assistente e assim definir um novo propósito.

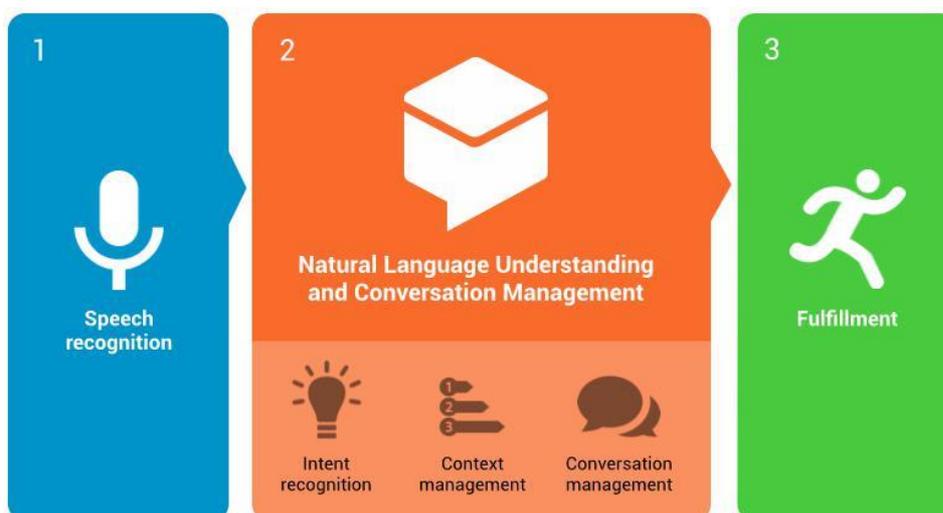
O uso de um NLP, que tem a mesma função do Natural Language Understander, citado anteriormente, facilita a tarefa de mapeamento de frases do usuário para funções discretas do software assistente, permitindo uma junção discreta entre o reconhecimento de voz e a execução de funções. É necessário configurar os padrões esperados pela expressões do usuário para facilitar o entendimento do NLP.

O Assistant.ai (API.AI, 2016a) da Api.ai (2016b) é um assistente de propósito geral. Tem como principal objetivo auxiliar o usuário a realizar tarefas comuns como gerenciar agendas, notas, tarefas, calendário e também responder questões gerais, e.g. questões relacionadas a dados sobre presidentes, países, pessoas famosas, cinema, notificar sobre eventos importantes, realizar traduções. Ele responde a questionamento do usuário, mas também sugere assistência baseada em informações de contexto, como indicar ações ao usuário baseado na combinação de informações registradas na agenda e na localização atual dele.

Sua arquitetura é multiplataforma e pode ser utilizado em dispositivos controlados pelos sistemas operacionais móveis Android (Google, 2016b), IOS (Apple, 2016b), ou que tenham o navegador web Chrome (Google, 2016d) instalado. Possui tecnologia de reconhecimento de linguagem natural com suporte a diversos idiomas, incluindo o inglês, o espanhol e o português.

Uma das características mais relevantes desse assistente é a plataforma conversacional em que ele foi projetado. Essa plataforma é composta por uma *Application Programming Interface* (API), chamada Api.ai (API.AI, 2016a), também fornecida pela empresa Api.ai, que é subdividida em três componentes modularizados: Speech Recognition, Natural Language Understanding e Conversation Management (CM), e User-Fulfillment, conforme Figura 8.

Figura 8 - API que suporta a Assistant.ai



Fonte: Api.ai (2016a)

O Natural Language Understanding, que possui responsabilidades semelhantes as de um processador de linguagem natural, e o Conversation Management fazem parte do principal componente da plataforma, responsável pela

gestão do diálogo com o usuário. O NLU suporta inserção de padrões de requisição do usuário, mapeamento de sinônimos, e customização de valores de referência para retorno.

Essa API é baseada na especificação Representational State Transfer (REST), a qual permite suprir o gerenciamento de estados para aplicações que utiliza comandos HTTP simples como POST, GET, PUT e DELETE. Esse recurso facilita a submissão de consultas, obtenção de resultados e o acionamentos de serviços que permitem, por exemplo, gerenciar completamente o conhecimento de um assistente. Através dessa API, é possível, por exemplo, criar assistentes que aprendem, i.e. ampliam o seu conhecimento por meio das informações inseridas pelos próprios usuários. Há disponíveis *Software Development Kits* (SDKs) e bibliotecas para os mais diversos sistemas operacionais como: Android, IOS, OS X (Apple, 2016c) e ambientes e linguagens de programação, e.g. HTML 5, Javascript, .NET, C++, Ruby, Python (PYTHON SOFTWARE FOUNDATION, 2016) entre outros.

Um dos assistente virtuais de maior utilização atualmente é o Siri, da Apple. Ele vem pré-instalado nos smartphones, da empresa desde de 2012. Atualmente já foram vendidos mais de um bilhão desses smartphones, e estima-se que o Siri esteja disponível em pelo menos 800 milhões desses equipamentos. O Siri é um assistente virtual de propósito geral que, segundo Gruber (2015), co-fundador da empresa que, originalmente o criou, foi lançado em 2010 e adquirido pela Apple no mesmo ano.

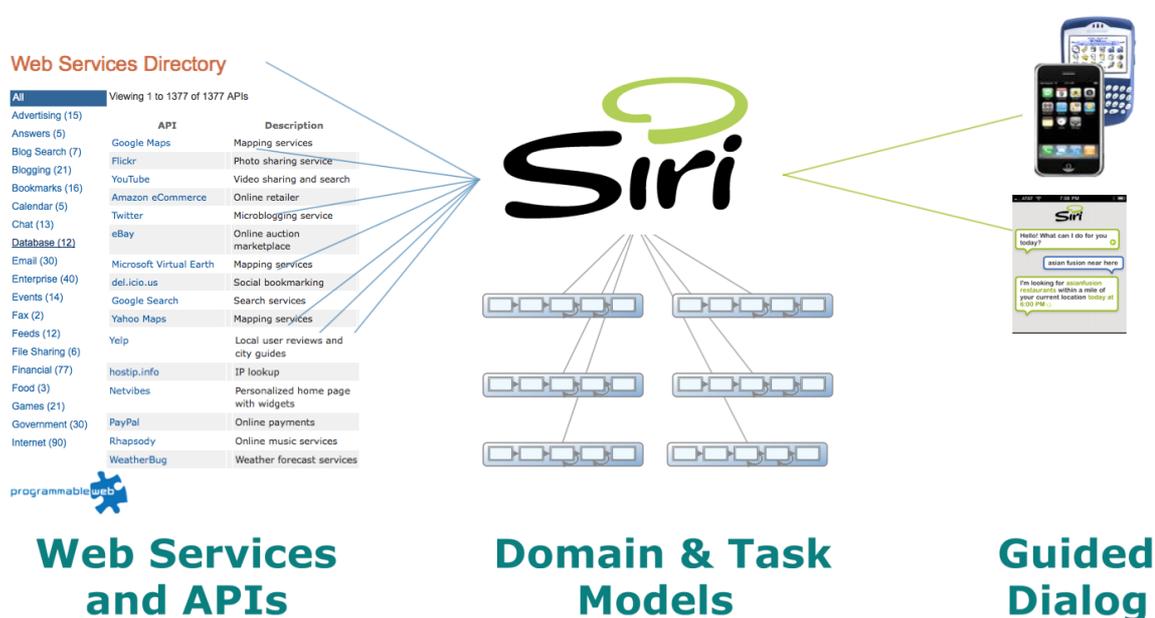
O propósito do Siri é auxiliar os usuários a realizar atividades comuns, e.g. planejar um jantar, encontrar uma loja, encontrar algo para fazer, planejar uma viagem, ir ao cinema. Para isso ele integra diversas tecnologias para funcionar, e.g. reconhecimento automático de voz, processamento de linguagem natural, detecção de intenções nas questões do usuário, tecnologias para combinação de diferentes aplicativos e serviços para se integrar com *web services*, serviços computacionais disponibilizados através da web, de outras empresas que realizam ações, pesquisas e respondem perguntas, conforme Figura 9. As respostas desses serviços são transformadas pelo Siri em uma linguagem natural e apresentadas por voz ao usuário, através da tecnologia de conversão de texto em voz.

O Siri manipula mais de uma dezena de domínios de conhecimento estruturados através da combinação de diversos *web services*. Estes serviços vão

sendo adicionados ao longo do tempo, ampliando a capacidade do conhecimento manipulado pelo Siri. Por se tratar de um software proprietário a sua arquitetura não está disponível para estudo e análise.

O Google Now (Google, 2016e) é um assistente virtual da Google, que auxilia os usuários, através de reconhecimento de voz e uso de linguagem natural, a encontrar diversas informações relacionadas ao contexto do usuário, e.g. informar o restaurante famoso mais próximo, informar o tempo estimado para chegar em algum local. Sua arquitetura também não está disponível.

Figura 9 – Siri – Tecnologia para integrar diversos serviços



Fonte: Gruber (2015)

## 2.7 CONSIDERAÇÕES FINAIS

Para a construção do Assistente Financeiro Virtual é necessário projetá-lo baseado em uma arquitetura de assistentes de propósito específico. Após um levantamento bibliográfico foram identificadas algumas propostas que permitem ou auxiliam a construção desse tipo de assistente.

O framework proposto por Martin et al (1996) provê a interação do usuário através de linguagem natural e voz. Uma das vantagens desse framework é o gerente de conversa, que permite que o usuário interaja com a aplicação com o uso de pronomes e/ou expressões com informações parciais, simulando um comportamento natural durante a conversa com o usuário. Entretanto, esse framework não é uma solução completa para construção de assistentes virtuais. O objetivo desse framework é o desenvolvimento de um componente intercambiável

que permita a uma aplicação interagir com o usuário através de linguagem natural por voz. Não está previsto pelo framework como a informação extraída da expressão do usuário será utilizada para executar alguma ação. Também não está previsto como será construída a resposta do usuário. Outra desvantagem é que a construção das regras da gramática é complexa. Além disso, esse framework foi projetado principalmente para aplicações que utilizavam o sistema de telefonia como interface com os usuários, a 20 anos atrás, e apresenta tecnologias que já estão ultrapassadas, i.e., conversão de texto em voz,

O framework proposto por Chihani et al. (2013) propõe que a construção do comportamento do assistente seja realizada pelo próprio usuário, de uma forma facilitada com o uso de linguagem natural. A assistência que será realizada é personalizada. Entretanto, o comportamento do assistente está limitado à disponibilidade e às funcionalidades das aplicações instaladas no dispositivo móvel. Outra limitação é que para invocar as aplicações instaladas no dispositivo o framework é dependente do mecanismo de comunicação interno do sistema operacional móvel Android. Essa característica limita a sua implementação para outros sistemas operacionais. Além disso, para esse framework o domínio do conhecimento está nas aplicações instaladas. Ele não oferece funcionalidades para construir o domínio, pois sua principal finalidade é combinar domínios disponibilizados pela aplicações.

A arquitetura proposta por Paraiso e Barthès (2005) é adequada para criação de assistentes de propósito específico. Ela permite a ampliação do conhecimento e do comportamento do assistente através da adição de novos serviços. Prevê a interação através de linguagem natural e suporta a gestão de diálogo com o apoio de ontologias. Entretanto ela foi projetada para implementação de assistentes na tecnologia de agentes, criada há mais de 18 anos por Lange (1998), e atualmente está em desuso, apesar da euforia inicial com o potencial da plataforma. Apesar de ser um trabalho de mais de onze anos, uma das contribuições dele para o desenvolvimento desta dissertação foi a arquitetura de interface com o usuário.

Uma das grandes desvantagens dessa arquitetura é a necessidade de um ambiente específico para os agentes funcionarem. Normalmente os *frameworks* e *containers* para agentes mais utilizados foram implementados em linguagem Java e muitos deles já foram descontinuados. Além disso, definir o Assistente na forma de

um agente, conforme Zambiasi (2012), limita a implementação para assistentes apenas como agentes.

A arquitetura de referência proposta por Eisman et al (2012) serve para criação de assistentes de propósito específico, mas foi elaborada para construção de assistentes de auxílio a navegação de um site web, gerando respostas mais próximas possível ao usuário tentando ajudá-lo a encontrar a página que contém a informação que o usuário procura. O objetivo da abordagem não é fornecer a resposta precisa e correta para o que foi perguntado, e sim, disponibilizar informações ou alternativas para que o próprio usuário encontre a informação que procura na página ou site onde ela reside.

Por não ser uma abordagem por serviços, o uso desta arquitetura para geração de assistentes de pergunta e resposta que permitam a ampliação do conhecimento através da inclusão de serviços de informação de terceiros, não é adequada, pois, para se obter o mesmo resultado sem a utilização da abordagem por serviços, seria necessária uma grande modificação na sua estrutura. Apesar disso, a arquitetura proposta pelos autores serviu de inspiração para o AFV proposto nesta dissertação, em razão da utilização de um processamento de linguagem natural para reconhecer as solicitações do usuário.

Já arquitetura de Zambiasi (2012), que é orientada a serviços e permite a criação de assistentes que podem ter seu conhecimento ampliado por serviços de terceiros, não fornece tratamento a interações com o usuário utilizando linguagem natural. A gestão do diálogo com usuário está limitada as *interfaces* das aplicações do usuário. Para esta dissertação a arquitetura proposta por Zambiasi (2012) contribui com a definição de requisitos gerais arquiteturais para implementação de um assistente e com a identificação de elementos principais do respectivo modelo de referência.

Apesar do assistente Sys Virtual Assistant ser de propósito geral ele permite ampliar o seu conhecimento através da adição de novos plugins, o que possibilita a criação e adição de um módulo com conhecimento financeiro embutido. Este recurso é suportado pela plataforma independente desenvolvida também pela Synthetic Intelligence Network, chamada Syn Engine (SYN, 2015a), que não exige um grande poder computacional para funcionar. Essa plataforma está disponível através de licenças de uso, inclusive uma versão gratuita. Algumas desvantagens do assistente é que o Sys Virtual Assistant suporta apenas um único idioma, o inglês, além disso,

os únicos ambientes de software compatíveis atualmente são o .Net e o Mono, limitando uma grande quantidade de potenciais usuários. Além disso, não está prevista na arquitetura desse Assistente como consultar os dados do domínio e como construir a resposta para o usuário.

A plataforma de conversação Api.ai suporta mais de 18 idiomas entre eles inglês, espanhol e português e pode ser integrado a alguns softwares de assistência proprietários existentes no mercado. Uma característica de grande importância dessa plataforma, que está implementada em forma de API, é a sua capacidade para se criar novos domínios, definir comportamentos e, portanto, criar assistentes de propósito específico, além de poder ser utilizada de forma gratuita com fins de pesquisa ou não comerciais.

Uma das desvantagens dessa API é que as funcionalidades mais avançadas só estão disponíveis na versão paga. Além disso, determinadas informações dos usuários, como localização, nomes da agenda, das aplicações, artistas e *tags* dos arquivos de mídia são enviadas para os servidores da empresa para criar os ambientes de contexto, o que pode infringir em questões relacionadas a privacidade da informação do usuário.

Conclui-se que estas arquiteturas, *frameworks* e plataformas analisados nesse capítulo não suportam, de forma completa, a criação do Assistente Financeiro Virtual que foi proposto no Capítulo 1.

### 3 ARQUITETURA DO ASSISTENTE FINANCEIRO VIRTUAL

É proposta uma arquitetura específica para o Assistente Virtual Financeiro, em razão de não se ter encontrado na revisão da bibliográfica uma arquitetura que permita construir um assistente de propósito específico do tipo pergunta e resposta baseada em serviços e com suporte a linguagem natural e gestão de diálogo.

A arquitetura do Assistente Financeiro Virtual foi projetada em um ambiente distribuído e orientado a serviços. Ela é composta por quatro camadas distintas categorizadas por função: Presentation, Orchestration, Understanding / Knowledge, e Data, conforme Figura 10.

Figura 10 - Arquitetura em camadas do Assistente Financeiro Virtual



Fonte: Próprio autor.

A camada Presentation é a camada de interface com o usuário. É responsável por receber e tratar as interações com o usuário e encaminhá-las para a camada Orchestration. Ela também é responsável por receber as informações enviadas pela camada Orchestration e apresentá-las para o usuário no formato adequado, que pode ser texto, imagem, animação ou voz. A usabilidade do assistente depende diretamente desta camada.

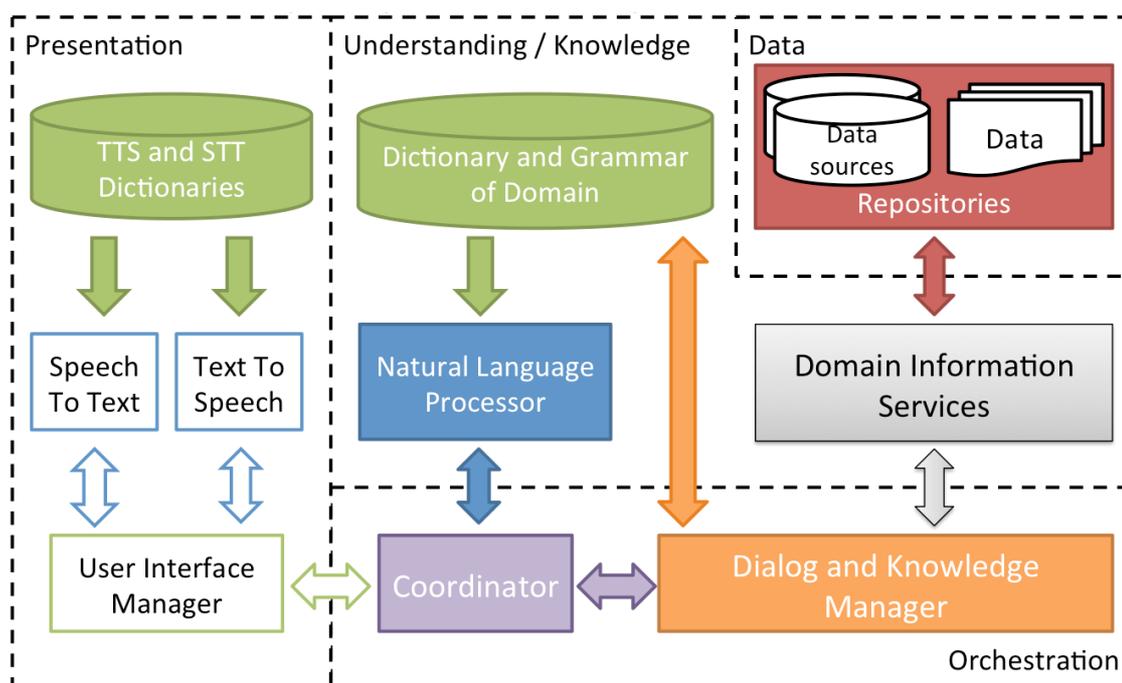
A camada Orchestration é a camada que coordena o assistente e faz a gestão do conhecimento e do diálogo com o usuário. Ela que define quais os serviços da camada Understanding / Knowledge serão acionados, e qual resposta deve ser encaminhada para a camada Presentation. Ela é a responsável por tomar a decisão do que deve ser feito em resposta aos estímulos das camadas Presentation e Understanding / Knowledge, como também, pelo tratamento de eventuais falhas dos serviços e/ou componentes. Ela também define se a informação passada pelo usuário está completa para gerar alguma resposta ou se é necessária alguma informação adicional a ser fornecida pelo usuário. A robustez do assistente depende diretamente desta camada.

A camada Understanding / Knowledge é o centro cognitivo do assistente. É ela quem interpreta as informações do usuário, fornece informações processadas para tomada de decisão e disponibiliza os serviços de informação especializada em determinado domínio de conhecimento para a camada Orchestration. A camada Understanding / Knowledge corresponde ao domínio do conhecimento do assistente, portanto, quanto maior representado o conhecimento estiver nessa camada, mais inteligente e conhecedor do domínio o assistente será. A sofisticação das respostas fornecidas pelo assistente depende diretamente do nível de sofisticação das informações disponibilizadas pelos serviços de informações especializadas embutidas na camada.

A camada *Data* tem como única responsabilidade fornecer dados para a camada Understanding / Knowledge. A confiabilidade das respostas do assistente é diretamente proporcional a confiabilidade dos dados fornecidos por esta camada. A quantidade de informações disponibilizada pelo assistente sofre influência direta da quantidade de dados a que esta camada tem acesso.

A arquitetura do assistente virtual de propósito específico segue o paradigma cliente-servidor. O lado cliente é representado pela camada Presentation. O lado servidor é formado pelas camadas Orchestration, Understanding / Knowledge e Data, conforme ilustrado na Figura 11.

Figura 11 - Arquitetura para assistentes virtuais de propósito específico.



Fonte: Próprio autor.

Conforme ilustrado na Figura 11, as camadas da arquitetura são compostas por serviços ou componentes. As setas indicam o fluxo de comunicação entre esses componentes. As camadas, que podem ser implementadas em diferentes tecnologias, serão descritas a seguir.

### 3.1 PRESENTATION LAYER

A camada Presentation é a camada de interface com o usuário, responsável por tratar as interações entre o assistente e o usuário. Ela é composta pelos componentes: User Interface Manager, Text To Speech, Speech To Text e os respectivos dicionários relacionados ao processamento da fala.

#### **3.1.1 Speech To Text**

O componente Speech To Text (STT) é um conversor de voz em texto, e, conforme planejado na arquitetura proposta, é o responsável por reconhecer as expressões faladas pelo usuário, normalmente encapsuladas em *streaming* de áudio, e convertê-las em texto. Normalmente acessa dicionários de palavra/voz para ampliar a sua precisão. Ele pode ser multilíngue, isto é, pode suportar a conversão de falas em texto em idiomas diferentes. Este componente depende diretamente de um dicionário específico para cada idioma de conversão. A precisão e a velocidade do assistente para o reconhecimento das expressões faladas pelo usuário são determinadas pela qualidade intrínseca do componente Speech To Text. Entretanto se não for obrigatório que o assistente reconheça as interações de voz do usuário, o componente Speech To Text é dispensável. Fatores como precisão e velocidade do reconhecimento da voz determinam a qualidade deste tipo de conversor.

#### **3.1.2 Text To Speech**

O componente Text to Speech (TTS) é um conversor de texto em voz, e, na arquitetura proposta, possui a responsabilidade de converter um texto em *streaming* de áudio de expressões faladas, processo inverso do componente Speech To Text. Semelhante ao componente Speech To Text ele geralmente acessa dicionários de palavra/voz para ampliar a precisão, também pode ser multilíngue. É dispensável caso não seja exigido que o assistente interaja por meio de áudio com o usuário.

Caso seja obrigatório, a pronúncia e a entonação da voz do assistente dependem diretamente da qualidade intrínseca do Text to Speech utilizado.

### **3.1.3 TTS and STT Dictionaries**

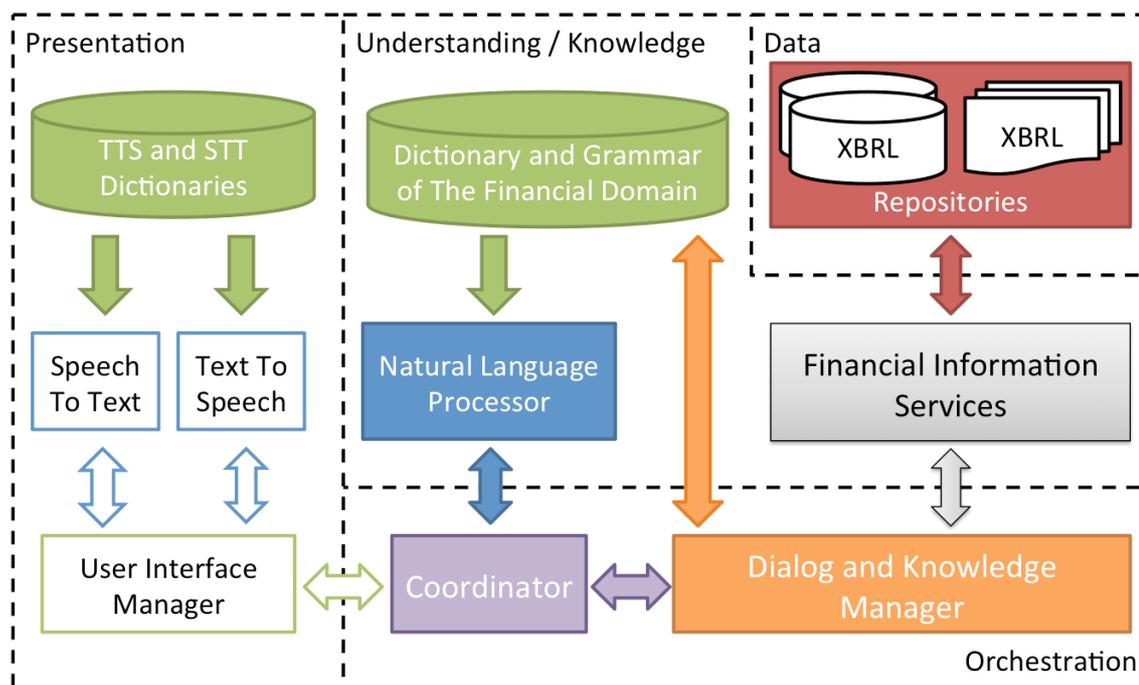
TTS and STT Dictionaries são bases de dados responsáveis por suportar as conversões de fala em texto, e vice-versa, para diferentes idiomas. Elas facilitam o processo de conversão nos componentes citados. Cada dicionário contém informações específicas para o algoritmo de conversão de determinado Speech To Text. ou Text to Speech. Quanto maior for a disponibilidade de dicionários de conversão em idiomas diferentes, maior será a capacidade multilíngue do assistente para interagir com o usuário.

### **3.1.4 User Interface Manager**

O componente User Interface Manager (UIM) é o responsável pela interação com o usuário, ele que coleta os questionamentos do usuário, encaminha-os para a camada Orchestration e aguarda a resposta. Ele também é que, eventualmente, interage por voz com o usuário através das devidas conversões prestadas pelos componentes Speech To Text e Text to Speech. A apresentação da resposta ao usuário também é de responsabilidade do UIM, e pode ser apresentada em diversos formatos como texto, gráfico, imagem ou voz, algo que depende dos recursos disponibilizados pelo hardware onde a aplicação cliente está sendo executada.

As camadas do lado servidor: Orchestration, Understanding / Knowledge e Data, são diretamente relacionadas ao domínio de conhecimento do assistente virtual. Os componentes e/ou serviços dessas camadas serão apresentados já representando o domínio financeiro, que é o domínio do Assistente Financeiro Virtual, a Figura 12 ilustra essa especialização na arquitetura proposta.

Figura 12 - Arquitetura do Assistente Financeiro Virtual



Fonte: Próprio autor

### 3.2 Orchestration Layer

A camada Orchestration é quem coordena todo o assistente, e gerencia o seu comportamento. Esta camada é a responsável por aguardar as requisições do usuário, encaminhadas pela camada Presentation, para reconhecimento e tratamento na camada Understanding / Knowledge. O tratamento consiste em verificar se todas as informações necessárias para construção de uma resposta foram fornecidas pelo usuário e requisitar serviços, disponíveis da camada Understanding / Knowledge, para construir a resposta do usuário. A camada Orchestration também gerencia o estado do diálogo entre o Assistente e o usuário, com o auxílio da camada Presentation. Ela é composta pelos componentes Coordinator e Dialog and Knowledge Manager (DKM), que serão apresentados a seguir.

#### 3.2.1 Coordinator

O componente Coordinator é o coordenador de todo o sistema, é o responsável por realizar diversas atividades básicas para que o Assistente funcione adequadamente. O Coordinator controla o ciclo de vida de cada transação iniciada pelo usuário e o fluxo de informação dentro do Assistente. Para auxiliar na execução

das tarefas, o Coordinator dispõe do contexto, ou estado, do Assistente, que contém todas as informações relacionadas à sessão de perguntas do usuário em andamento.

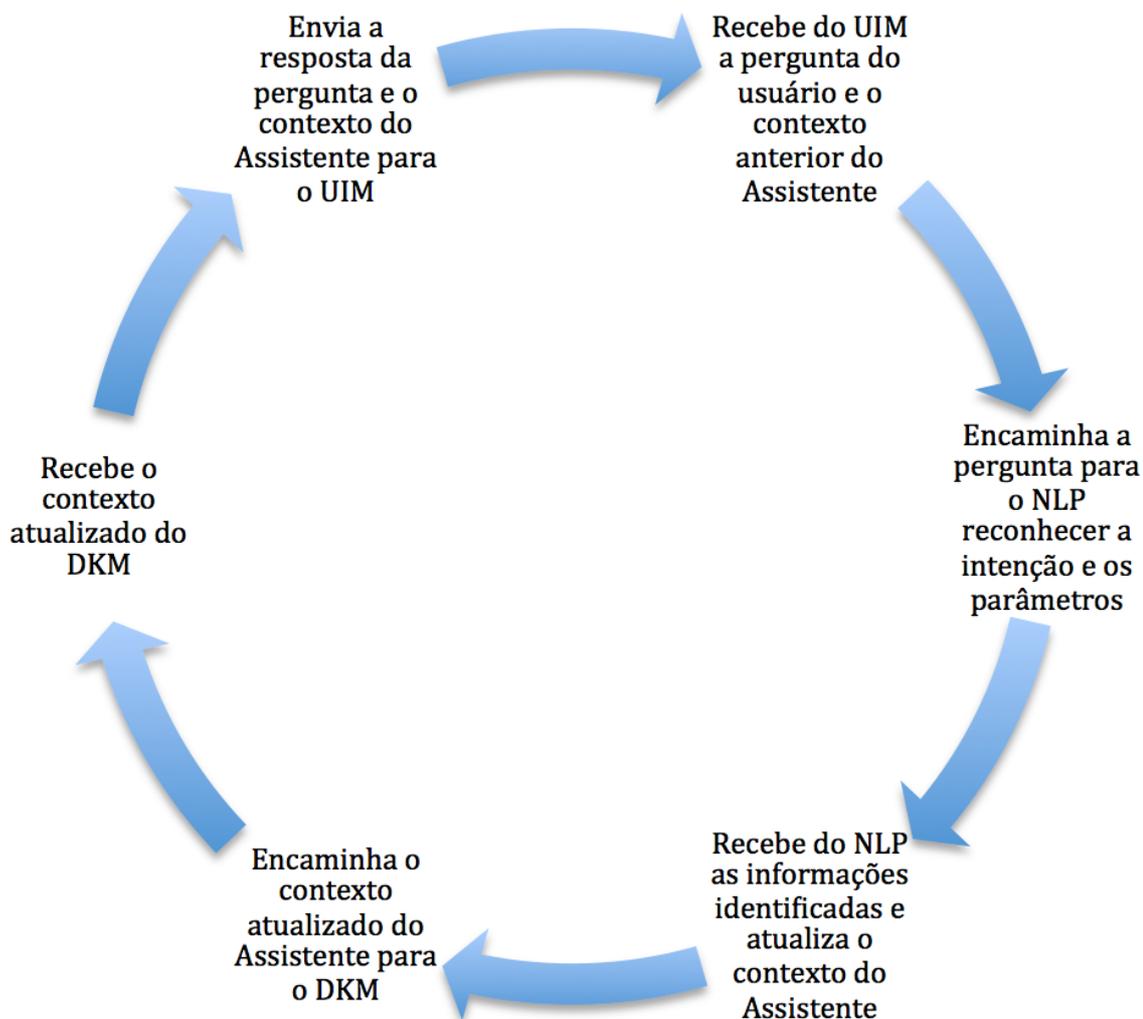
O Coordinator é o responsável por coletar as requisições do usuário, com os respectivos contextos, enviadas pelo User Interface Manager, e encaminhar o texto da pergunta, para o Natural Language Processor (NLP). Ele também tem a responsabilidade de atualizar o contexto com as informações referentes a intenção e os parâmetros reconhecido pelo NLP no texto do usuário.

Além disso, o Coordinator é o responsável por interagir com Dialog and Knowledge Manager, através de atualizações de contexto, para obter a resposta do usuário, e encaminhá-la ao User Interface Manager, bem como, possuir a responsabilidade de tratar as falhas das camadas com as quais interage e informar ao usuário sobre as eventuais ocorrências de falha, de forma amigável, também com o auxílio do User Interface Manager.

Para um melhor entendimento do funcionamento do Assistente, será apresentado o ciclo de vida de uma transação típica controlada pelo Coordinator, ilustrado na Figura 13, e o respectivo fluxo de informação.

Uma transação típica se inicia quando o Coordinator recebe um texto, que é a pergunta do usuário, e o contexto anterior do assistente, enviados através do User Interface Manager, da camada Presentation. O Coordinator atualiza o contexto do assistente, e, em seguida, encaminha o texto para o Natural Language Processor, da camada Understanding / Knowledge. Após receber como resposta a identificação da intenção e os parâmetros da pergunta do usuário, o Coordinator atualiza o contexto do assistente e o envia para o Dialog and Knowledge Manager. Ao receber do DKM o contexto atualizado, que contém a resposta da pergunta, o Coordinator envia para o usuário a resposta a ser apresentada e o contexto do assistente atualizado, através do User Interface Manager. Em seguida, ele finaliza a transação e aguarda pela próxima solicitação do usuário.

Figura 13 - Ciclo de vida de uma transação típica controlada pelo Coordinator.



Fonte: Próprio autor.

Nos casos em que o Natural Language Processor sinaliza que não reconheceu a intenção do usuário ou que tenha ocorrido algum erro, o Coordinator não aciona o Dialog and Knowledge Manager, ele encaminha uma mensagem de erro ao usuário como resposta para o User Interface Manager.

### **3.2.2 Dialog and Knowledge Manager**

O componente Dialog and Knowledge Manager é o gestor do diálogo com o usuário e do conhecimento do Assistente. Para que o diálogo aconteça de forma semelhante a uma conversa entre pessoas, o DKM utiliza o contexto do Assistente como suporte a manutenção do diálogo. A gestão do conhecimento, feita pelo DKM, consiste na orquestração e manutenção dos serviços do domínio de conhecimento presentes na camada Understanding / Knowledge.

Para um melhor entendimento desse processo, será detalhada a participação do Dialog and Knowledge Manager no ciclo de vida, de uma transação típica. Avaliando o fluxo da informação, apresentado anteriormente e ilustrado na Figura 13, após o Dialog and Knowledge Manager receber do Coordinator o contexto do Assistente, que já contém a nova intenção e os novos parâmetros identificados, ele analisa diversas situações, e.g. se o usuário elaborou uma nova pergunta completa, se o usuário alterou uma parte da pergunta feita anteriormente, se o usuário confirma uma pergunta feita pelo assistente ou se o usuário selecionou um item em uma lista informada anteriormente pelo Assistente. Além disso o DKM verifica se recebeu todos parâmetros necessários para acionar o Financial Information Service (FIS) indicado no contexto, em seguida, encaminha os parâmetros e a identificação do Financial Information Service selecionado ao serviço, ou componente, Financial Information Services. Após receber a mensagem do Financial Information Services, o DKM atualiza o contexto e o encaminha para o Coordinator.

O Dialog and Knowledge Manager também atualiza as informações de contexto do assistente referentes ao último questionamento do usuário permitindo manter um diálogo em situações onde as questões subsequentes são incompletas ou baseadas na pergunta anterior. Esta funcionalidade tem como objetivo simular o comportamento de manutenção de uma conversa natural com o usuário. Nessa conversa, por exemplo, após responder com sucesso uma pergunta do usuário, com a ajuda do contexto o assistente tenta responder as perguntas subsequentes relacionadas a primeira pergunta, mesmo que elas sejam incompletas.

O Dialog and Knowledge Manager, em conjunto com Financial Information Services, permite que o Assistente incorpore novos conhecimentos ou novos comportamentos, i.e. incluir novos serviços de informação financeira. A adição de novos padrões ou comportamentos pode ser realizada dinamicamente.

### 3.3 UNDERSTANDING / KNOWLEDGE LAYER

A camada Understanding / Knowledge é a camada responsável pela representação do domínio do conhecimento e pelo entendimento das questões elaboradas pelo usuário relacionadas ao domínio. Ela é acionada exclusivamente pela camada Orchestration e fornece todas as informações necessária para a construção da resposta. A camada é composta pelos componentes: Natural Language Processor, que trata as sentenças típicas do domínio do conhecimento

elaboradas em linguagem natural; Grammar of The Financial Domain, que define as regras de sintaxe para as perguntas financeiras do usuário; Dictionary of The Financial Domain, que contém o vocabulário financeiro reconhecido pelo Assistente; Financial Information Services, que corresponde um conjunto organizado de serviços que fornecem informações financeiras especializadas.

### **3.3.1 Natural Language Processor**

O serviço Natural Language Processor possui como principal responsabilidade reconhecer a sentença do usuário, elaborada em linguagem natural, e extrair dela informações parametrizadas que permitam a um sistema computacional executar procedimentos em resposta a mesma. Para manipular o conhecimento o NLP é dependente do dicionário e da gramática de domínio, isto é, ao receber uma expressão para processamento, ele os consulta a fim de entender o que o usuário sentenciou.

Um dos pontos chaves desse processo é a extração dos parâmetros da expressão realizada em linguagem natural. É importante salientar que por ser um assistente de propósito específico ele não reconhecerá qualquer solicitação do usuário. Ele somente conseguirá reconhecer solicitações que contenham termos do dicionário e que correspondam aos padrões de pergunta previamente definidos na gramática do Natural Language Processor. Caso não encontre um padrão coerente com a pergunta realizada, o NLP sinaliza para o Coordinator que não reconheceu a expressão do usuário. Em caso positivo, a intenção do usuário e os parâmetros reconhecidos no processo de entendimento são encaminhados para o Coordinator para o devido tratamento.

### **3.3.2 Dictionary of The Financial Domain**

O componente Dictionary of The Financial Domain é o dicionário que define quais os termos, ou palavras, e o seus respectivos sinônimos, do vocabulário do domínio financeiro são reconhecidos pelo Assistente Financeiro Virtual.

Antes de aprofundar sobre qual a função, ou qual a responsabilidade, do Dictionary of The Financial Domain na arquitetura proposta, serão destacados alguns conceitos linguísticos.

O significado de “palavra” corresponde a uma:

Unidade que pertence a uma das grandes classes gramaticais, como substantivo (quando expressa um objeto), verbo (quando expressa uma ação), adjetivo (quando expressa uma qualidade), preposição (quando expressa relação) etc. considerando apenas seu significado, sem levar em conta as modificações que nela ocorrem com as marcações flexionais; vocábulo. (MELHORAMENTOS, 2016).

Já o significado de dicionário corresponde a uma:

Coleção, parcial ou completa, das unidades lexicais de uma língua (palavras, locuções, afixos etc.), em geral dispostos em ordem alfabética, com ou sem significação equivalente, assim como sinônimos, antônimos, classe gramatical, etimologia etc., na mesma ou em outra língua. (MELHORAMENTOS, 2016).

Por extensão o dicionário significa uma:

Compilação de informações ou referência a respeito de um tema (ou área específica de um ramo do conhecimento) ordenada de forma alfabética aos moldes de um dicionário; glossário, vocabulário: dicionário da construção. (MELHORAMENTOS, 2016).

As palavras chaves relacionadas a área do domínio financeiro, como nomes de documentos, índices e conceitos financeiros, que o Assistente Financeiro Virtual tem que reconhecer, obrigatoriamente devem estar representadas nesse dicionário. Além disso, verbos e preposições típicas do domínio financeiro que se quer que o Assistente reconheça, também devem ser registradas no dicionário para facilitar a correta identificação da intenção do usuário embutida na pergunta.

Os significados dos termos do dicionário são valores que serão encaminhados como parte dos parâmetros de consulta dos serviços de informação financeira disponíveis e, por isso, deverá haver um conhecimento prévio dos valores esperados por esses serviços, a fim de construir um dicionário compatível.

Para assistentes multilíngue é necessária a criação de dicionários específicos para cada idioma, pois os vocábulos e seus respectivos significados são distintos. Quanto mais representativo for cada dicionário, mais fácil será para o Assistente reconhecer as palavras típicas do domínio no idioma desejado.

### **3.3.3 Grammar of The Financial Domain**

O componente Grammar of The Financial Domain, que corresponde a gramática do Assistente Financeiro Virtual, define a sintaxe das questões financeiras elaboradas pelo usuário que o processador de linguagem natural conseguirá reconhecer.

Antes de detalhar o que é este componente, é importante conhecer o significado de “gramática”, que é:

Conforme a linguística descritiva, exposição sistemática e objetiva dos elementos constitutivos (fonemas, morfemas, palavras, frases etc.), dos processos de construção e dos recursos expressivos próprios do sistema estrutural de uma dada língua. (MELHORAMENTOS, 2016).

Como é uma gramática de apoio ao processador de linguagem natural, ela consiste em padrões de perguntas que podem ser feitas pelo usuário, onde cada padrão indica ao processador de linguagem natural em quais posições os parâmetros esperados devem ser extraídos. No caso do Assistente Financeiro Virtual, as formas de indagação típicas do jargão financeiro devem ser registradas nessa gramática.

Além disso, em linguagem natural uma mesma pergunta pode ser realizada de diversas formas, ou padrões, e estas diversas formas devem ser previstas na gramática para que o processador reconheça a maior quantidade possível delas, e, assim, conseguir simular uma maior naturalidade do Assistente em reconhecer com correção o que está sendo perguntado. Por exemplo, em português, as seguintes perguntas tem o mesmo significado: “Qual o lucro da empresa X em 2014?” e “Em 2014, qual o lucro da empresa X?”, e, portanto, os respectivos padrões devem ser registrados na gramática para reconhecimento correto delas pelo Assistente.

É importante destacar que, para que o assistente reconheça com sucesso uma quantidade maior de expressões do usuário, a gramática deve ser elaborada de forma integrada com o dicionário, i.e. quanto mais o domínio financeiro estiver representado no dicionário e na gramática, maior será a capacidade do Natural Language Processor em reconhecer os questionamentos financeiros, e, por conseguinte, maior será a precisão do assistente para entender o que o usuário está perguntando em relação ao domínio.

Para assistentes multilíngue é necessária a existência de um par dicionário/gramática específico para cada idioma alvo, pois, geralmente, a sintaxe, e a maioria dos vocábulos, de um idioma são diferente de outros, o que diminui a possibilidade de utilizar um mesmo par dicionário/gramática para idiomas distintos.

### **3.3.4 The Financial Information Services**

Os serviços Financial Information Services são os responsáveis por tratar os dados disponíveis na camada Data e fornecer informações do domínio financeiro solicitadas pela camada Orchestration. Eles representam o domínio de conhecimento do assistente, isto é, quanto maior for abrangência dos serviços disponíveis na área de conhecimento maior será a representatividade do conhecimento do Assistente. A quantidade e qualidade das análises disponibilizadas pelo Assistente dependem dos serviços de informações que ele acessa.

Baseado na disponibilidade desses serviços, o Assistente Financeiro Virtual pode fornecer diversas informações financeiras, e.g. disponibilizar dados do balanço financeiro, informar o significado de determinados termos financeiros, expor o resultado do cálculo de indicadores financeiros, ou até mesmo disponibilizar o resultado da análise de riscos financeiros de uma determinada empresa.

Estes serviços de informações financeiras podem estar em qualquer local e podem ser implementados nas mais diversas tecnologias. Entretanto, uma das características primordiais que eles devem possuir é a velocidade no retorno das consultas, pois as respostas do Assistente devem ser realizadas em poucos segundos.

O conjunto de Financial Information Services possui o mapeamento entre as intenções do usuário e os respectivos serviços de informação financeira, que fornecerão as respostas referentes aos parâmetros identificados.

A capacidade do assistente em responder perguntas complexas, como análises e comparações financeiras, depende diretamente da disponibilidade de serviços que fornecem estas respostas. i.e, a quantidade, a versatilidade e a complexidade das informações financeiras disponibilizadas pelos serviços de informação financeiras são os principais responsáveis pelo nível de sofisticação das respostas fornecidas pelo Assistente Financeiro Virtual.

## **3.4 DATA LAYER**

A camada Data é a responsável por fornecer os dados para camada Understanding / Knowledge. Ela é composta por repositórios ou bancos de dados ou outras fontes de dados, e, no caso do Assistente Financeiro Virtual, a camada Data está representada pelos XBRL Repositories. A quantidade de dados disponibilizados

por estes repositórios, e a taxonomia que eles utilizam, são alguns dos fatores limitantes do alcance das respostas do assistente. Além disso, a correção dos dados da camada Data influencia diretamente o Assistente Financeiro Virtual em fornecer informações corretas em suas respostas.

### **3.4.1 XBRL Repositories**

Os serviços ou componentes XBRL Repositories disponibilizam dados financeiros para a camada Understanding / Knowledge. São os responsáveis por suprir as informações contábeis para análise financeira. Eles podem ser formados por documentos XBRL, por bancos de dados XBRL ou serviços de informações financeiras baseados em dados em XBRL. É importante que os repositórios utilizados pelo Assistente Financeiro Virtual sejam mantidos periodicamente por instituições idôneas o que corroborará para que as respostas fornecidas sejam baseadas em dados atuais e autênticos.

A taxonomia utilizada pelo repositório XBRL é também um item que interfere na aderência das respostas do Assistente Financeiro Virtual à área financeira. É necessária que essa taxonomia tenha representatividade no domínio financeiro, que ela tenha amplitude de uso na comunidade financeira e que, de preferência, ela seja administrada por órgão ou entidade de referência no mercado financeiro. Em sincronismo com as gramáticas e dicionários do Natural Language Processor, é possível ocorrer o reconhecimento simultâneo de termos, ou palavras, pertencentes a taxonomias distintas

## **3.5 FLUXO DE INFORMAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL**

Para propiciar uma compreensão mais clara do funcionamento integrado dos componentes, serviços e camadas da arquitetura utilizada, será apresentada uma descrição do fluxo de informação do Assistente Financeiro Virtual.

Para uma pergunta do usuário, feita por voz, cujo o padrão seja de conhecimento do Assistente, o fluxo de informação mínimo necessário para respondê-la, também através de voz, é composto por 16 passos principais, conforme detalhado a seguir:

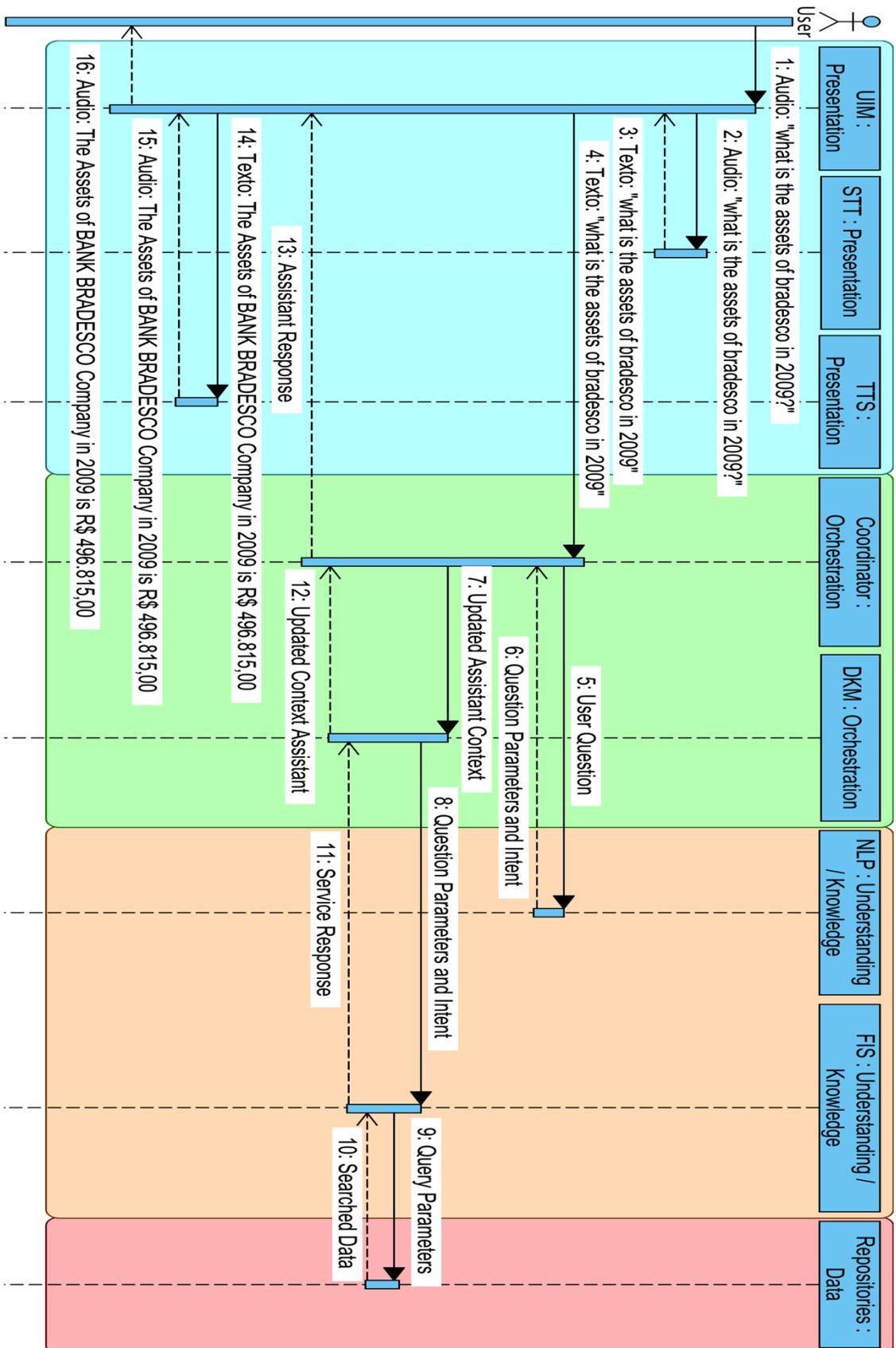
- 1.O usuário faz uma pergunta através da voz e ela é capturada pelo User Interface Manager, da camada Presentation;

- 2.O User Interface Manager encaminha o stream de áudio, que corresponde a pergunta do usuário, para o Speech To Text, da camada Presentation;
- 3.O Speech To Text retorna o texto, correspondente ao stream de áudio, para o User Interface Manager;
- 4.O User Interface Manager encaminha o texto, da pergunta do usuário, para o Coordinator, da camada Orchestration;
- 5.O Coordinator encaminha o texto do usuário para a Natural Language Processor, da camada Understanding / Knowledge;
- 6.O Natural Language Processor retorna a intenção do usuário e os parâmetros reconhecidos no texto do usuário;
- 7.O contexto atual do Assistente, no qual foram embutidos a intenção e os parâmetros reconhecidos no texto, é encaminhado para o Dialog e Knowledge Manager, da camada Orchestration;
- 8.O Dialog e Knowledge Manager analisa o contexto do Assistente e encaminha a intenção e os parâmetros para o Financial Information Services, da camada Understanding / Knowledge;
- 9.O serviço, ou componente, Financial Information Services, baseado na intenção, seleciona um Financial Information Service e encaminha para ele os parâmetros enviados pelo DKM. O Financial Information Service, selecionado, realiza uma query em um serviço de dados de um repositório, da camada Data;
- 10.O serviço de dados retorna os valores da consulta em formato próprio para o Financial Information Service;
- 11.O Financial Information Service constrói a resposta com os valores recebidos e retorna para o Dialog e Knowledge Manager;
- 12.O Dialog e Knowledge Manager atualiza o Contexto do Assistente com a resposta do Financial Information Service e retorna o Contexto atualizado para o Coordinator;
- 13.O Coordinator constrói a resposta do Assistente Financeiro Virtual com o Contexto recebido e retorna a resposta para o User Interface Manager;
- 14.O User Interface Manager extrai da resposta recebida o texto a ser falado pelo Assistente e encaminha o texto para o Text To Speech;
- 15.O Text To Speech converte o texto em stream de áudio e retorna o stream para o User Interface Manager;

16.Finalmente, o User Interface Manager fornece a resposta da pergunta ao usuário por meio de áudio.

A Figura 14 ilustra um exemplo do diagrama de sequência dos passos da resposta, por voz, do Assistente Financeiro Virtual à pergunta “What is the assets of bradesco in 2009?” submetida, também por voz, pelo usuário.

Figura 14 - Diagrama de Sequência de uma consulta ao Assistente Financeiro Virtual



Fonte: Próprio autor.

### 3.6 CONSIDERAÇÕES FINAIS

Uma das principais vantagens da arquitetura apresentada é a previsão da utilização de serviços, componentes, de processamento de linguagem natural, permitindo a substituição do fornecedor do serviço, ou componente, ou até mesmo, a utilização de mais de um serviço de processamento de linguagem natural. Essa característica facilita a implementação de assistentes virtuais de propósito específico multilíngues, ou de assistentes que reconheçam o vocabulário de domínios distintos. Facilita também a seleção de serviços de processamento de linguagem natural de domínios específicos do conhecimento que estejam disponíveis no mercado, ou na comunidade acadêmica, permitindo a criação de assistentes virtuais multi-domínios.

A arquitetura possibilita que o Assistente atue com domínios de conhecimento distintos simultaneamente, além de prever a adição de novos termos de domínio de forma dinâmica.

O isolamento das camadas responsáveis pelo domínio do conhecimento e cognição do Assistente, das outras camadas, e a previsão de uso de serviços de conversão de voz, na camada de apresentação, se caracterizam como outra vantagem da arquitetura proposta, pois possibilitam a criação de assistentes virtuais que interagem com seus usuários exclusivamente através de voz, sem a exigência de codificações complexas na implementação da interface com usuário.

Uma das limitações dessa arquitetura é que ela não suporta a implementação de qualquer tipo de assistente de propósito específico, i. e. ela foi projetada para construir assistentes virtuais do tipo pergunta e resposta. Assistentes virtuais autônomos, que, conforme alguma situação, autonomamente realiza alguma tarefa, ou uma sequência de tarefas, não são suportados por essa arquitetura.

No próximo capítulo será apresentado como o Assistente Financeiro Virtual foi implementado.

#### 4 IMPLEMENTAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL

O objetivo principal da Implementação é validar o Assistente Financeiro Virtual, e viabilizar os testes e avaliações para responder as questões de pesquisa propostas neste trabalho.

Buscou-se uma configuração com baixo acoplamento e que permitisse modificar o comportamento do assistente em um só lugar para todos os clientes, isto proporciona maior facilidade de manutenção do mapeamento em um único artefato de software, trazendo vantagens como rapidez na intervenção, menor probabilidade de falhas de atualização de códigos, menor quantidade de testes, menor custo de alteração.

Implementou-se quase todas as camadas em ambiente servidor, com exceção da camada Presentation que teve um dos seus componentes implementados no ambiente cliente.

Implementou-se os seguintes artefatos de software:

- uma aplicação cliente móvel em Sistema Operacional (SO) Android;
- uma aplicação cliente em html / javascript para ser executado em navegadores web;
- uma aplicação web em Java, representando a camada Orchestration e seus componentes;
- *web services* em Java representando os serviços de informação financeira;
- um *web service proxy* em Python para auxiliar no acesso a um serviço do Natural Language Processor proprietário utilizado neste trabalho.

A tecnologia web foi selecionada para ser a base do Assistente Financeiro Virtual, principalmente pela disponibilidade dela nos sistemas operacionais mais utilizados no mercado, e.g. Linux, OS X, Unix e Windows.

A aplicação e os serviços web foram implementados na tecnologia Java (ORACLE, 2016) em razão da disponibilidade da plataforma para os mais diversos sistemas operacionais e da existência de diversas ferramentas do tipo *Integrated Development Environment (IDE)*, ou Ambiente Integrado de Desenvolvimento, disponíveis no mercado, algumas delas gratuitas, e.g., NetBeans (ORACLE, 2016b)

e Eclipse (THE ECLIPSE FOUNDATION, 2016). IDE é um programa de computador que reúne características e funcionalidades para facilitar o desenvolvimento de software.

Além disso a plataforma Java é amplamente conhecida por profissionais do mercado e pela academia, o que facilita a integração do Assistente Financeiro Virtual com outras bibliotecas em Java já implementadas.

O Sistema Operacional móvel Android foi escolhido para ser o ambiente da aplicação web cliente em razão de alguns fatores, como: gratuidade para desenvolver e distribuir aplicativos; possuir a maior quantidade de usuários de sistemas móveis do mundo; vir instalado em quase todos os smartphones e *tablets* de custo baixo; utilizar o ambiente Java para criação de aplicações; disponibilidade de diversos IDEs no mercado, alguns deles gratuitos, e.g. Android Studio (GOOGLE, 2016c) e Eclipse (THE ECLIPSE FOUNDATION, 2016).

Outro fator importante é que os smartphones e *tablets* com o SO Android comumente possuem microfone, auto-falante e bibliotecas de software embutidas, característica que facilita avaliar a execução do Assistente Financeiro Virtual quando submetido a perguntas feitas pelo usuário através da voz.

Com o objetivo de permitir a utilização do Assistente Financeiro Virtual nos mais diversos sistemas operacionais, implementou-se uma segunda aplicação em linguagem javascript para ser executada em navegador web. Todos os navegadores web disponíveis nos sistemas operacionais, seja ele para ambiente móvel ou para ambiente desktop, possuem um interpretador javascript.

O uso da linguagem Python e do recurso de proxy deve-se ao serviço de processador de linguagem natural selecionado para este trabalho. Para utilizar o serviço o fornecedor do processador definiu alguns procedimentos, entre eles, um de autenticação de requisição. Esse procedimento é realizado por funcionalidades embutidas em bibliotecas de software disponibilizadas pelo próprio fornecedor, entretanto, até a conclusão da implementação do Assistente Financeiro Virtual, nenhuma biblioteca Java estava disponível. Como havia uma das versões dessa biblioteca em Python, construiu-se uma aplicação proxy do serviço em um servidor web com suporte a Python para obter e repassar informações entre a aplicação web Coordinator e o serviço de processador de linguagem natural. Os critérios da seleção do serviço de processador de linguagem natural serão apresentados no Capítulo 5.

A seguir serão descritas as implementações dos serviços, ou componentes, de cada camada que compõe Assistente Financeiro Virtual.

#### 4.1 IMPLEMENTAÇÃO DA PRESENTATION LAYER

O componente User Interface Manager é o *front-end*, a interface visual e interativa do assistente financeiro virtual. É ele que interage com o usuário capturando as perguntas dele e respondendo-as. Este foi o único componente da camada Presentation implementado em razão de não haver soluções disponíveis e por ser específico para o Assistente Virtual Financeiro. Atualmente os demais componentes da camada são disponibilizados nas mais diversas formas e por várias empresas de desenvolvimento de software, seja através de *plugins* em navegadores web, ou por meio de serviços. Normalmente eles suportam diversos idiomas, mas dificilmente identificam o idioma automaticamente, necessitando de alguma configuração do usuário. Para aumentar a eficiência no reconhecimento e na execução da fala, alguns desses componentes acionam serviços na Internet para concluir o processamento.

As implementações dessa camada serão apresentadas por meio de exemplos.

##### **4.1.1 Aplicação cliente móvel**

Para o Assistente Virtual Financeiro implementou-se uma aplicação móvel, conforme Figura 15, compatível com o sistema operacional Android com o objetivo de facilitar a avaliação de todos os componentes da camada Presentation. Esta facilidade acontece porque o Android disponibiliza nativamente os componentes Text To Speech, Speech To Text e os respectivos dicionários para diversos idiomas, entre eles o inglês e o português. Além disso, todos os smartphones possuem microfone e alto falante, imprescindíveis para testar e avaliar a interação por voz com o assistente. Apesar da implementação ter utilizado os componentes de voz da Google, é possível utilizar os componentes ou serviços de outros fornecedores.

Figura 15 - Telas da aplicação móvel do Assistente Financeiro Virtual



Fonte: Próprio autor.

Na tela inicial da aplicação móvel o usuário pode falar a pergunta ou digitá-la. Para submeter a pergunta por voz, basta o usuário digitar no ícone, que aparece na tela, e falar a pergunta seguida de uma pausa. Para submeter por texto, basta digitar a pergunta no teclado virtual que aparece quando o campo de entrada, sublinhado em vermelho, está selecionado, e digitar a tecla “enter”. A resposta aparecerá no campo de Resposta sempre acima das últimas respostas apresentadas. A última pergunta submetida pelo usuário ao Assistente aparece em um campo correspondente, facilitando ao usuário reutilizá-la totalmente, ou em parte. Na Figura 15 é possível visualizar a situação inicial da tela do Assistente e situação da tela após o usuário submeter a pergunta “Qual o ativo da Petrobras em 2009?”.

#### **4.1.2 Aplicação cliente em página HTML/Javascript**

Codificou-se também uma outra versão do User Interface Manager, uma aplicação em página web utilizando linguagem html e script javascript, com o objetivo de permitir a execução da interface do Assistente Financeiro Virtual nos navegadores web mais utilizados, ampliando a possibilidade de uso nos mais

diversos sistemas operacionais disponíveis. Essa versão permite ao usuário elaborar perguntas apenas através do teclado do dispositivo.

O funcionamento é semelhante a versão móvel. Para submeter a pergunta é necessário digitar a pergunta no campo de entrada e clicar no botão “Submeter pergunta”, ou acionar a tecla “enter”. A última pergunta submetida pelo usuário ao Assistente aparece em um campo nomeado “Pergunta realizada”. A resposta aparecerá abaixo de todas as outras informações. Na Figura 16 é possível visualizar duas situações distintas da página html / javascript do Assistente. A primeira situação é quando a página é carregada, e a segunda ocorre após o usuário submeter a pergunta “Qual o ativo da Petrobras em 2009?”.

Figura 16 – Telas da aplicação do Assistente Financeiro Virtual em HTML/Javascript

**Assistente Financeiro Virtual**

Português  English

Faça uma pergunta financeira sobre alguma empresa

Submeter pergunta

---

**Assistente Financeiro Virtual**

Português  English

Faça uma pergunta financeira sobre alguma empresa

Qual o ativo da Petrobras em 2009

Submeter pergunta

Pergunta realizada:  
Qual o ativo da Petrobras em 2009

O(A) Ativo da empresa PETROBRAS - PETROLEO BRASILEIRO SA no periodo de 2009 corresponde a US\$200,270.00

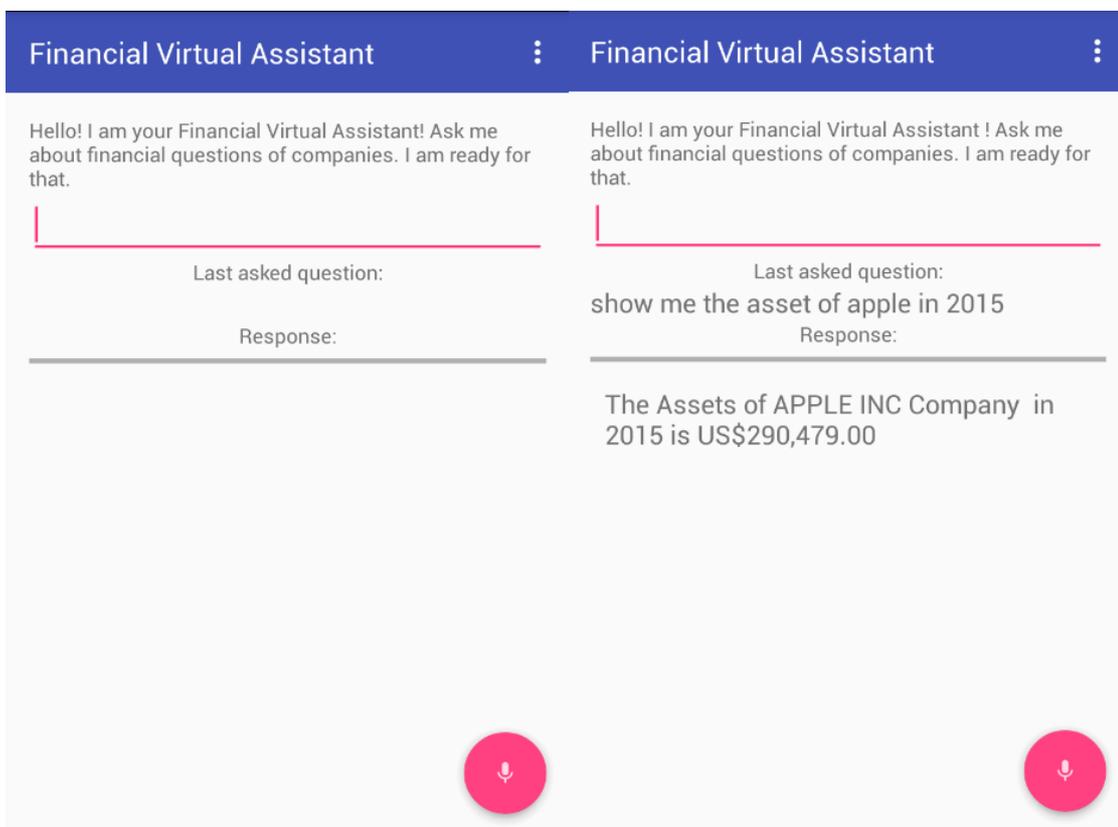
Fonte: Próprio autor.

### 4.1.3 Tratamento multilíngue dos clientes

As duas versões clientes foram implementadas para suportar o idioma inglês e o idioma português. As mensagens para o usuário foram inseridas no código nos dois idiomas. A versão móvel foi implementada para apresentar e tratar as informações no idioma que está configurado no dispositivo do usuário.

Para que o processamento de voz ocorra corretamente, a aplicação seleciona os componentes Speech To Text e Text To Speech de acordo com o idioma configurado. Caso o dispositivo não possua os componentes correspondentes instalados, a aplicação informa ao usuário que o dispositivo não suporta o recurso de voz para o idioma em configurado. A Figura 17 mostra a situação da tela inicial e a situação da tela após o Assistente Financeiro Virtual responder a pergunta “show me the asset of apple in 2015” submetida pelo usuário.

Figura 17 - Telas em inglês da versão móvel do Assistente Financeiro Virtual



Fonte: Próprio autor.

Já para a versão em página HTML / Javascript é necessário que o usuário selecione qual o idioma o Assistente Financeiro Virtual deve processar. Quando o usuário clica na opção do idioma desejado, as informações aparecem no idioma

correspondente. Na Figura 18 mostra as telas no idioma inglês dessa versão, a primeira tela é a tela inicial da aplicação e a segunda corresponde a tela de resposta à pergunta “Show me The assets of Apple in 2015” que o usuário submeteu ao Assistente Financeiro Virtual.

Figura 18 – Telas em inglês da versão HTML / Javascript do Assistente Financeiro Virtual

**Financial Virtual Assistant**

Português  English

Ask a financial question about some company

Query submit

---

**Financial Virtual Assistant**

Português  English

Ask a financial question about some company

Show me The assets of Apple in 2015

Query submit

Asked Question:  
Show me The assets of Apple in 2015

The Assets of APPLE INC Company in 2015 is US\$290,479.00  
Overall Execution Time: 3491.694999999924 milliseconds.

Fonte: Próprio autor.

## 4.2 IMPLEMENTAÇÃO DA ORCHESTRATION LAYER

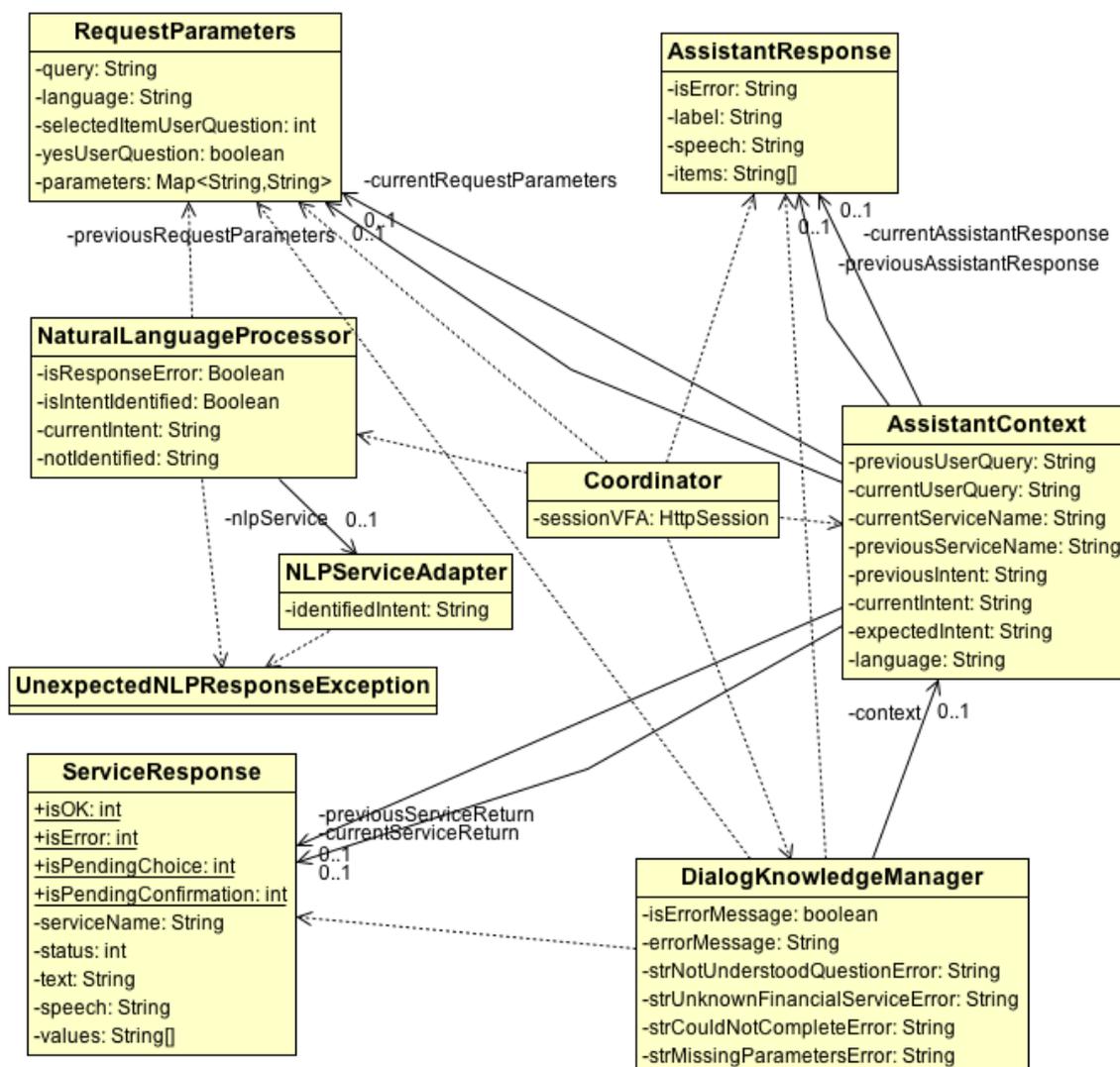
Implementou-se os componentes da camada Orchestration utilizando tecnologia Java para execução em um servidor web. Isso permitirá que os componentes tratem as requisições de cliente web, possibilitando que qualquer usuário que possua um navegador web utilize o Assistente.

Diversas classes Java foram implementadas, entretanto, as seguintes classes são fundamentais para que a camada Orchestration exerça a sua função de maestro do Assistente Financeiro Virtual:

- A classe Coordinator que representa o componente Coordinator, tem a função é de controlar o fluxo de informações e coordenar o acionamento dos componentes e serviços básicos do Assistente.
- A classe AssistantContext que corresponde as informações do contexto do Assistente em relação a sessão de diálogo com o usuário;
- A classe RequestParameters, que armazena todas as informações relacionadas aos parâmetros da pergunta do usuário;
- As classes NaturalLanguageProcessor e NLPServiceAdapter, que são as representantes do Natural Language Processor;
- A classe AssistantResponse que corresponde a resposta do Assistente à pergunta do usuário;
- A classe DialogKnowledgeManager que representa o componente Dialog and Knowledge Manager, cuja responsabilidade é gerenciar o conhecimento do Assistente e o diálogo entre o Assistente e o usuário;
- A classe ServiceResponse que corresponde a resposta do Serviço de informação à query submetida.

Para auxiliar no entendimento da relação entre as classes implementadas, está ilustrado na Figura 19 o diagrama de classes da implementação da camada Orchestration.

Figura 19 - Diagrama de classes da implementação da camada Orchestration



Fonte: Próprio autor.

#### 4.2.1 Classe Coordinator

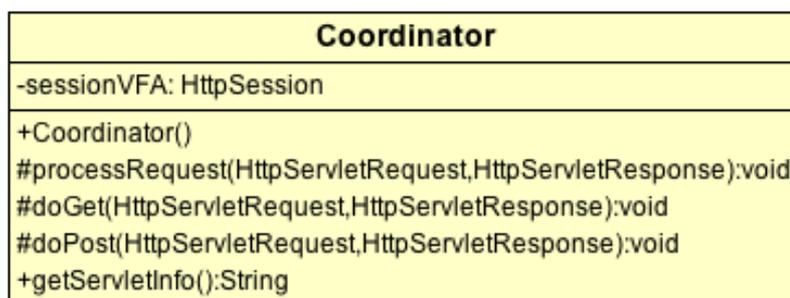
Implementou-se o componente Coordinator através de uma classe, nomeada Coordinator, que implementa a classe abstrata HttpServlet da tecnologia Java. Ao estender a classe HttpServlet, a classe Coordinator, ilustrada na Figura 20, pode ser executada em um servidor web e tratar requisições de clientes web utilizando o protocolo http.

Os métodos doGet, doPost, herdados da super classe HttpServlet, são invocados quando um cliente web submete uma requisição para o endereço do Assistente Financeiro Virtual. Eles foram sobrecarregados para direcionar as requisições http, GET e POST, para o método processRequest, evitando tratar de

forma diferente estes tipos de requisição http. O método `processRequest` é onde está o algoritmo principal da classe `Coordinator`.

A classe `Coordinator` utiliza as classes `AssistantContext`, `AssistantResponse`, `DialogKnowledgeManager`, `NaturalLanguageProcessor` e `RequestParameters` para realizar as atividades operacionais do Assistente Financeiro Virtual, como: receber e tratar as requisições web do usuário e a variável de sessão do contexto do Assistente, encaminhar o texto submetido para o processador de linguagem natural, encaminhar o resultado do processamento para o gestor de diálogo e enviar a resposta do Assistente para o cliente web do usuário.

Figura 20 - Classe `Coordinator`



Fonte: Próprio autor.

A classe `Coordinator` não manipula nenhuma informação relacionada ao domínio financeiro, isto é, ela não acessa nenhuma das classes do serviço financeiro para executar suas ações operacionais, conforme o diagrama de classes do Assistente Financeiro Virtual, ilustrado na Figura 19.

#### **4.2.2 Classe `AssistantContext`**

A classe `AssistantContext`, ilustrada na Figura 21, implementa o contexto do Assistente Financeiro Virtual e tem a responsabilidade de armazenar todas as informações necessárias relacionadas a manutenção da sessão de perguntas do usuário.

Figura 21 - Classe AssistantContext

<b>AssistantContext</b>
-previousUserQuery: String -currentUserQuery: String -currentRequestParameters: RequestParameters -previousRequestParameters: RequestParameters -currentServiceName: String -previousServiceName: String -previousIntent: String -currentIntent: String -expectedIntent: String -language: String -previousServiceReturn: ServiceResponse -currentServiceReturn: ServiceResponse -previousAssistantResponse: AssistantResponse -currentAssistantResponse: AssistantResponse
+AssistantContext() +getPreviousUserQuery():String +setPreviousUserQuery(String):void +getCurrentUserQuery():String +setCurrentUserQuery(String):void +getCurrentRequestParameters():RequestParameters +setCurrentRequestParameters(RequestParameters):void +getPreviousRequestParameters():RequestParameters +setPreviousRequestParameters(RequestParameters):void +getCurrentServiceName():String +setCurrentServiceName(String):void +getPreviousServiceName():String +setPreviousServiceName(String):void +getPreviousIntent():String +setPreviousIntent(String):void +getCurrentIntent():String +setCurrentIntent(String):void +getExpectedIntent():String +setExpectedIntent(String):void +getLanguage():String +setLanguage(String):void +getPreviousServiceReturn():ServiceResponse +setPreviousServiceReturn(ServiceResponse):void +getPreviousAssistantResponse():AssistantResponse +setPreviousAssistantResponse(AssistantResponse):void +getCurrentServiceReturn():ServiceResponse +setCurrentServiceReturn(ServiceResponse):void +getCurrentAssistantResponse():AssistantResponse +setCurrentAssistantResponse(AssistantResponse):void

Fonte: Próprio autor.

- a) Estas informações correspondem a:
- b) Perguntas, atual e anterior, do usuário;
- c) Parâmetros reconhecidos na pergunta atual e na pergunta anterior;
- d) Nome do serviço a ser acionado e o nome do serviço acionado anteriormente;
- e) Intenção atual e anterior do usuário, além da intenção do usuário esperada pelo Assistente;
- f) Idioma atual;
- g) Resposta do serviço anterior e resposta do serviço atual;
- h) Resposta atual do assistente e a resposta anterior.

Todas as informações registradas na classe `AssistantContext` são exclusivas para a manipulação do contexto do Assistente e nenhuma delas tem relação com classes do domínio financeiro. Essa classe é fundamental para as ações realizadas pelas classes `Coordinator` e `DialogKnowledgeManager`, que a utilizam para obter as informações do contexto do Assistente, as quais servem de base para tomada de decisão.

#### **4.2.3 Classe *RequestParameters***

Os parâmetros identificados no texto da requisição do usuário são encapsulados em um objeto da classe `RequestParameters`, ilustrada na Figura 22, e são utilizados por quase todas as classes da camada `Orchestration`, conforme o diagrama de classes ilustrado na Figura 19. A classe `RequestParameters` tem a responsabilidade de guardar as informações relacionadas aos parâmetros da consulta do usuário, tanto os parâmetros identificados pelo serviço de NLP, quanto os parâmetros relacionados ao idioma e ao tipo de pergunta. Estas informações auxiliam na construção das respostas, na confirmação de alguma operação, na seleção de um item em uma lista, e na identificação do idioma utilizado na pergunta.

A classe `RequestParameters` não possui nenhuma dependência com classes ou funcionalidades de nenhum `Natural Language Processor` específico. Os parâmetros recebidos por qualquer NLP, são armazenados em pares (nome do parâmetro e o seu valor) na variável `parameters`, que é uma implementação da interface `Map`, da tecnologia Java.

A interface Map é uma representação de uma tabela hash, que é uma estrutura de dados que faz associações entre chaves de pesquisa e seus respectivos valores. Esta interface estipula que não podem existir chaves duplicadas em um mapeamento. As justificativas para utilizar uma classe que implementa a classe Map são: a facilidade de uso para encontrar o valor de um parâmetro no mapeamento, a disponibilidade de implementação deste tipo de estrutura de dados nas mais diversas linguagens de programação, e a reduzida quantidade de parâmetros que serão armazenados nela.

Figura 22 - Classe RequestParameters

<b>RequestParameters</b>
-query: String -language: String -selectedItemUserQuestion: int -yesUserQuestion: boolean -parameters: Map<String,String>
+RequestParameters() +getQuery():String +setQuery(String):void +getLanguage():String +setLanguage(String):void +getSelectedItemUserQuestion():int +setSelectedItemUserQuestion(int):void +getYesUserQuestion():boolean +setYesUserQuestion(boolean):void +getParameters():Map<String,String> +setParameters(Map<String,String>):void

Fonte: Próprio autor.

A classe RequestParameters não possui nenhuma dependência com classes relacionadas ao domínio de conhecimento, isto é, ela não é uma classe especializada em parâmetros relacionados a informação financeira. Ela pode ser utilizada para armazenar parâmetros de qualquer domínio manipulado pelo Assistente.

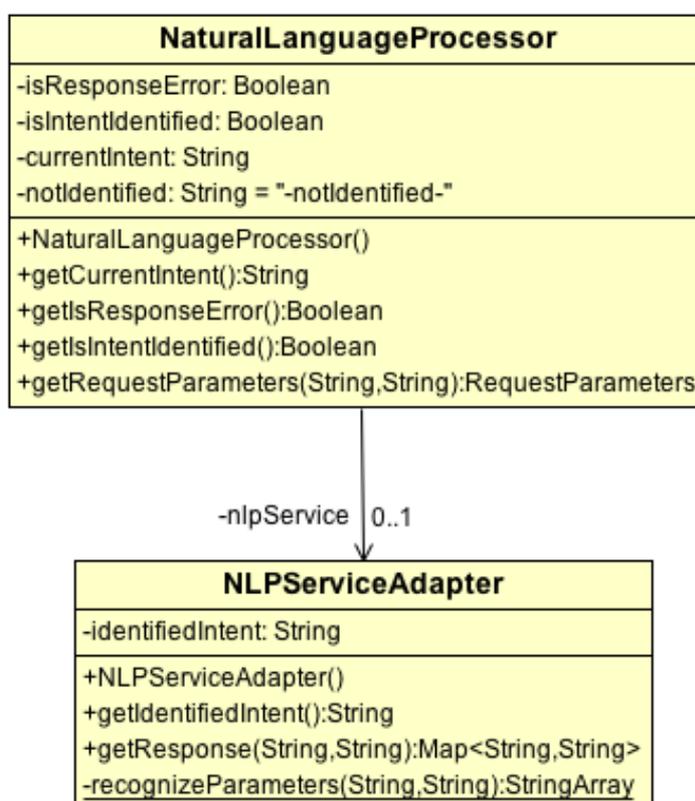
#### **4.2.4 Classes *NaturalLanguageProcessor* e *NLPServiceAdapter***

Criou-se as classes *NaturalLanguageProcessor* e *NLPServiceAdapter*, conforme o diagrama de classes ilustrado na Figura 23, com o objetivo de desacoplar a especificidade de um serviço, ou componente, responsável pelo processamento de linguagem natural, do restante da implementação do código do

Assistente, facilitando a utilização de outros componentes, ou serviços, de processamento de linguagem natural disponíveis no mercado.

A classe `NaturalLanguageProcessor` é a representante do processador de linguagem natural do Assistente. A sua função é retornar as informações resultantes do processamento do texto enviado pelo usuário, como: a intenção, ou tipo, da pergunta do usuário; os nomes dos parâmetros e os respectivos valores identificados, além de informar se, após o processamento do texto, a intenção do usuário foi reconhecida e se houve algum erro na execução.

Figura 23 - Diagrama das classes `NaturalLanguageProcessor` e `NLPServiceAdapter`



Fonte: próprio autor

A classe `Coordinator`, já descrita na Seção 3.2.1, encaminha o idioma e o texto do usuário para a classe `NaturalLanguageProcessor` através do método `getRequestParameters` e, como resultado, recebe os parâmetros e valores identificados, embutidos no objeto da classe `RequestParameters`, descrita na seção 4.2.3. A classe `NaturalLanguageProcessor` é a responsável por encapsular os dados identificados no processamento da linguagem natural na classe `RequestParameters`. Para verificar se a intenção foi identificada com sucesso ou se

houve algum erro com o processamento, a classe `Coordinator` consulta os métodos `getIsIntentIdentified` e `getIsResponseError` da classe `NaturalLanguageProcessor`.

A classe `NLPServiceAdapter`, descrita do diagrama de classes ilustrado na Figura 23, atua como um adaptador, exclusivo, do serviço de processamento de linguagem natural, fornecido pela empresa `Api.ai`, para a classe `NaturalLanguageProcessor`. A justificativa pela utilização deste serviço da `Api.ai` neste trabalho será apresentada na Seção 4.3, que apresenta a implementação da camada `Understanding / Knowledge`.

A implementação das duas classes, `NaturalLanguageProcessor` e `NLPServiceAdapter`, para realizar o processamento de linguagem natural tem como objetivo facilitar o desacoplamento da implementação do `Natural Language Processor`, que pode ser realizado por um componente ou por um serviço, do restante do código do Assistente. Conforme o diagrama de classes da camada `Orchestration`, ilustrado na Figura 19, a classe `Coordinator` não acessa a classe `NLPServiceAdapter`, ela somente acessa a classe `NaturalLanguageProcessor`.

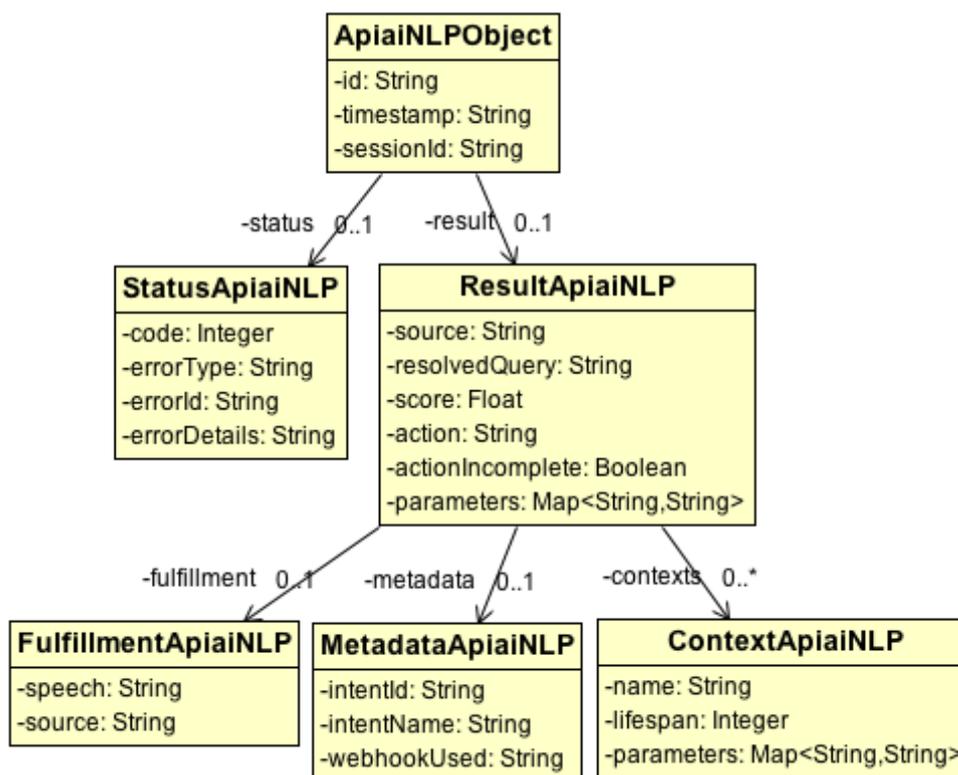
Quando a classe `NaturalLanguageProcessor` submete o texto e o idioma do usuário através dos métodos `getResponse` da classe `NLPServiceAdapter`, a classe `NLPServiceAdapter` encaminha as informações para o serviço do NLP da `Api.ai`, através do seu método, privado e estático, `recognizeParameters`. Este método é que acessa o *web service* do NLP da `Api.ai` e obtém o objeto de resposta do referido serviço, que é em um formato proprietário. Em seguida, a classe `NLPServiceAdapter` converte este objeto de resposta, da `Api.ai`, em uma implementação da *interface* `Map`, que corresponde a uma estrutura de dados do tipo tabela *hash*, e encaminha como resposta para a classe `NaturalLanguageProcessor`.

O objeto de resposta do NLP da `Api.ai` é enviado em formato texto, e utiliza uma codificação que segue a notação `JavaScript Object Notation (JSON)` (ECMA INTERNATIONAL, 2013), que é um formato não proprietário, independente de linguagem de programação e adequado para intercâmbio de dados. Ao receber este objeto `JSON`, a classe `NLPServiceAdapter` tem a tarefa de convertê-lo em uma instância de uma classe `Java`, a fim de facilitar o acesso a informação armazenada e evitar manipular informações em arquivo texto. Entretanto, para a tecnologia `Java` essa recriação não é trivial, pois consiste em criar e instanciar uma classe, que ainda não existe em tempo de execução, baseada em uma informação registrada em texto. A dificuldade nesse processo acontece porque a linguagem `Java` é

fortemente “tipada”, i.e. uma linguagem em que os tipos das variáveis devem ser declaradas, ou conhecidas em tempo de compilação, característica que impede a transformação de um texto em notação JSON, que é objeto *String*, diretamente para instâncias de classes que não foram declaradas em tempo de compilação, diferente da linguagem Javascript, por exemplo, onde a transformação de uma variável em tipos diferentes pode ser feita dinamicamente.

Para que a conversão fosse possível, foi necessário implementar a classe *ApiaiNLPObjeto*, conforme diagrama de classes ilustrado na Figura 24, que é compatível com a notação JSON do objeto de resposta proprietário do serviço de processamento de linguagem natural da *Api.ai*.

Figura 24 - Diagrama de classes do objeto de resposta do NLP da *Api.ai*



Fonte: próprio autor

Além disso, como nos kits de desenvolvimento de software padrão Java disponíveis não possuem bibliotecas que fazem a conversão e geração dinâmica de objetos JSON em uma instância de uma classe, para este trabalho foi necessária a utilização de uma biblioteca externa aos kits. Após uma seleção, utilizou-se uma biblioteca Java chamada Jackson (JACKSON PROJECT HOME, 2016). Ela foi escolhida por ser de código aberto, popular e por existir diversos materiais de orientação de uso dela disponíveis na Internet.

#### 4.2.5 Classe *AssistantResponse*

A classe *AssistantResponse* foi criada para encapsular as respostas do assistente e disponibilizar mais recursos, do que apenas uma mensagem de texto, para as interfaces das aplicações cliente. A classe *AssistantResponse*, ilustrada na Figura 25, disponibiliza o texto da mensagem da resposta, um texto indicando se é uma mensagem de erro, uma mensagem para ser falada quando pertinente pelo dispositivo do usuário e também uma eventual lista de itens ao usuário.

Figura 25 - Classe *AssistantResponse*

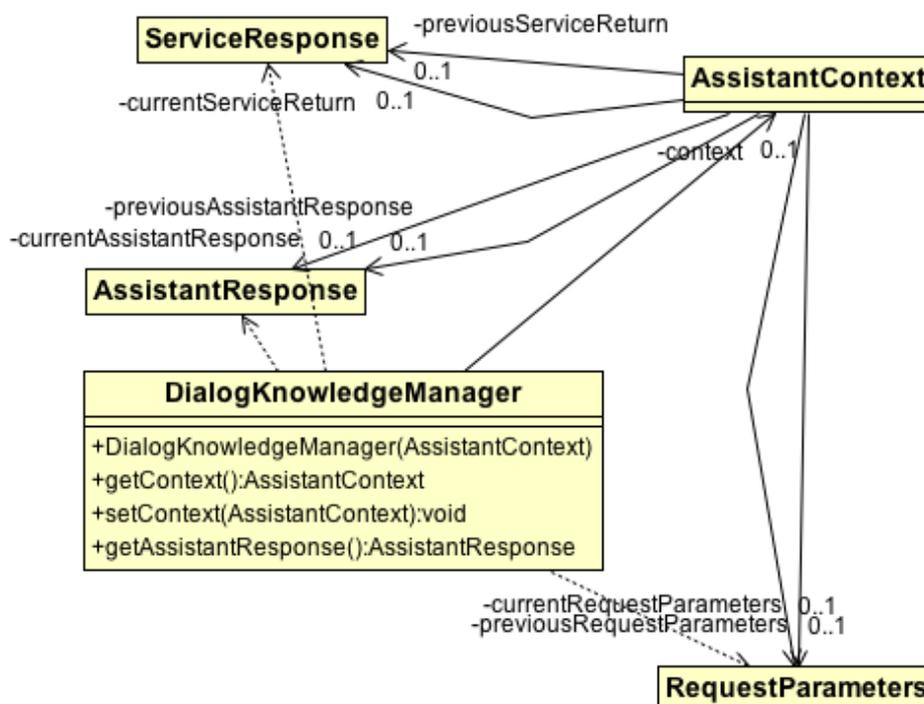
<b>AssistantResponse</b>
-isError: String
-label: String
-speech: String
-items: String[]
+AssistantResponse()
+getIsError():String
+setIsError(String):void
+getLabel():String
+setLabel(String):void
+getSpeech():String
+setSpeech(String):void
+getItems():String[]
+setItems(String[]):void

Fonte: Próprio autor

#### 4.2.6 Classe *DialogKnowledgeManager*

A classe *DialogKnowledgeManager* é a implementação do componente *Dialog and Knowledge Manager*, que é o responsável por gerir o diálogo do Assistente com o usuário, e por administrar o conhecimento do Assistente. O acionamento de serviços de informação financeira também é sua responsabilidade. Para realizar estas ações ela acessa as classes *AssistantContext*, *AssistantResponse*, *RequestParameters* e *ServiceResponse* de acordo com o diagrama de classes ilustrado na Figura 26.

Figura 26 – Diagrama de classes relacionadas a classe DialogKnowledgeManager



Fonte: Próprio autor.

A gestão do diálogo entre o Assistente e o usuário é realizada com o apoio da classe AssistantContext. Através do método getCurrentIntent da classe AssistantContext, a DialogKnowledgeManager verifica se a intenção do usuário corresponde a perguntas incompletas, i.e. perguntas curtas que levam em conta informações passadas em perguntas realizadas anteriormente. Quando a DialogKnowledgeManager encontra esse tipo de intenção ele atribui aos parâmetros que não foram identificados na pergunta atual, os valores dos parâmetros correspondentes identificados na pergunta anterior.

As intenções incompletas capturadas são:

- changePeriod,
- changeCompany,
- changePeriod,
- changeFinancialConcept,
- changeFinancialConceptAndPeriod,
- changeCompanyAndPeriod,
- changeCompanyAndFinancialConcept.

Para esclarecer melhor como a classe `DialogKnowledgeManager` atua, será analisado o seguinte exemplo de uma sequência de duas perguntas feitas pelo usuário, sendo que a primeira pergunta, que é completa, foi processada pelo Assistente e a segunda pergunta, que é incompleta, pois só informa a empresa, tem os respectivos parâmetros e intenção identificados registrados no contexto do Assistente, conforme a seguir:

- 1.“Qual o ativo da Vale em 2009?”
- 2.“E o da Petrobras?”

Para esse exemplo, após a segunda pergunta, no contexto do Assistente o valor da intenção corresponderá a `changeCompany`. Ao obter esse valor a classe `DialogKnowledgeManager` entende que o usuário somente informou a empresa pois ele quer utilizar a mesma pergunta realizada anteriormente com a empresa informada. A intenção de cada pergunta é identificada pelo Natural Language Processor e como essa identificação ocorre, está apresentada na Seção 4.3.4 .

Neste caso a classe `DialogKnowledgeManager` obtém todos os valores dos parâmetros informados na pergunta anterior, com exceção do parâmetro `Company`, e os atribui aos correspondentes parâmetros atuais, que estão sem valor. O valor da intenção anterior é atribuído ao valor da intenção atual e em seguida a classe `DialogKnowledgeManager` submete estes dados para o serviço `Financial Information Services`. Esses dados submetidos equivalem a pergunta “Qual o ativo da Petrobras em 2009?”

Quando a classe `DialogKnowledgeManager` certifica que todos os parâmetros necessários para acionar o serviço de informação financeira indicado pela intenção do usuário já são conhecidos, ela faz uma chamada ao método `getServiceResponse` do *web service* `FinancialInformationServicesSwitch`, implementado para este trabalho, que será apresentado na Seção 4.3.5 .

Para esta chamada a classe `DialogKnowledgeManager` encaminha duas *strings* como parâmetros da consulta do método `getServiceResponse`. A primeira *string* corresponde a intenção do usuário e a segunda corresponde a instância da classe `RequestParameters`, que contém os parâmetros identificados na pergunta do usuário, encapsulada em formato JSON.

O texto seguinte é uma continuação do exemplo citado anteriormente, no instante em que a classe `DialogKnowledgeManager` converte uma instância da classe `RequestParameters` em formato JSON. Essa instância corresponde a:

```
{"query":"Qual o ativo da petrobras em 2019","language":"pt-br","selectedItemUserQuestion":0,"yesUserQuestion":false,"parameters":{"yearPeriod":"","financialConcept":{"name":"assets","label":"Ativo"},"year":"2009","companyData":{"name":"PETROBRAS - PETROLEO BRASILEIRO SAI","cik":"0001119639","ticker":"pbr","stockExchange":"NYSE"}}}
```

O JSON facilita a conversão de instâncias de classes em formato texto, e consequentemente, o intercâmbio dessas instâncias entre serviços e objetos de tecnologias distintas.

A classe `DialogKnowledgeManager` recebe como resposta do serviço `FinancialInformationServicesSwitch` um texto em JSON, que corresponde uma instância da classe `ServiceResponse`, que será apresentada na Seção 4.2.7. Após recriar a instância, a classe `DialogKnowledgeManager` atualiza o contexto do Assistente com essa instância e retorna o contexto para o `Coordinator`, apresentado na Seção 3.2.1.

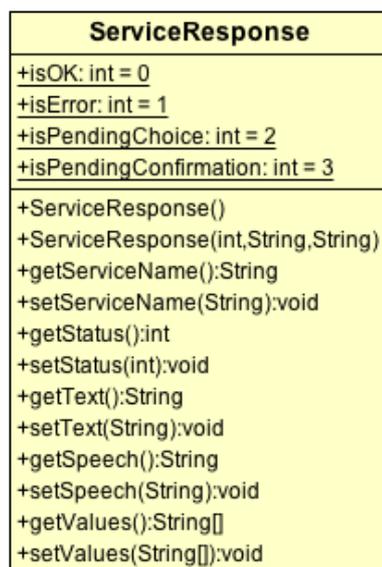
Caso a resposta do serviço `FinancialInformationServicesSwitch` seja uma string vazia, a classe `DialogKnowledgeManager` criará uma instância da classe `ServiceResponse`, com uma mensagem informando que ocorreu algum problema com o Assistente e não foi possível concluir o processamento, e a colocará no contexto, para posterior encaminhamento para a classe `Coordinator`.

#### **4.2.7 Classe Service Response**

A classe `ServiceResponse`, ilustrada na Figura 27, representa a resposta do serviço de informação financeira. Nessa resposta, além da mensagem de retorno do serviço, outras informações são registradas, e.g.: texto da mensagem a ser falada pelo assistente, uma lista de valores e uma meta-informação relacionado ao significado do estado do serviço. O estado do serviço pode corresponder a situações

como: ocorreu algum erro de conclusão do serviço; o serviço aguarda confirmação ou uma informação complementar; o serviço foi executado com sucesso.

Figura 27 - Classe ServiceResponse



Fonte: próprio autor

Todo Financial Information Service deve retornar como resposta uma instância da classe ServiceResponse.

### 4.3 Implementação da Understanding / Knowledge Layer

A camada Understanding / Knowledge foi implementada por serviços. Para o Natural Language Processor foi selecionado um serviço disponível no mercado, já os Financial Information Services foram implementados especificamente para o Assistente Financeiro Virtual.

#### **4.3.1 Natural Language Processor**

O Natural Language Processor é o responsável por reconhecer os parâmetros e intenções do usuário embutidas nas perguntas elaboradas pelo usuário em linguagem natural. Para facilitar a implementação do Assistente Financeiro Virtual, optou-se por não implementar um Natural Language Processor em razão de existir diversos serviços de processamento de linguagem natural disponíveis atualmente no mercado. Além disso, a implementação de um NLP envolve uma grande complexidade que extrapola o escopo deste trabalho.

Alguns serviços de Natural Language Processor são proprietários, e cada um possui características diversas que podem ser adequadas ou não para um assistente de propósito específico. Para a seleção do serviço de NLP do Assistente Financeiro Virtual os seguintes critérios foram avaliados, conforme mostrado no Quadro 2.

1. Dispõe de funções / API para criar novos domínios – Característica que permite a criação do domínio financeiro;
2. Suporta agrupamento de padrões de pergunta e customização de parâmetros – Funcionalidade que facilita o mapeamento das intenções do usuário aos serviços financeiros correspondentes;
3. É multilíngue – Pode atuar com mais de um idioma;
4. Disponibiliza licença gratuita para avaliação e pesquisa – O seu uso não onera o custo da realização do trabalho;
5. Está disponível para diversas plataformas – Característica que proporciona flexibilidade tecnológica para implementação, além de ampliar a base de usuários do Assistente;
6. Permite a manutenção da gramática e do dicionário de termos sem a necessidade de recompilar o código da aplicação – Característica que permite a construção de assistentes com conhecimento ampliável dinamicamente;
7. É um serviço ou disponibiliza as suas funcionalidades como tal – Característica que facilita a implementação do Assistente de forma compatível com a arquitetura.

Apesar de todos os Natural Language Processors avaliados serem multilíngues, nem todos suportavam o português. Três dos NLPs avaliados, Api.ai, Nuance Mix e Syn Engine, possibilitam a criação de novos domínios de conhecimento, como o domínio financeiro para o Assistente, conforme mostrado no Quadro 1. O NLP Nuance Mix não pode ser avaliado porque até a fase de seleção dos NLPs, o seu fornecedor, a Nuance Communications Inc, não havia disponibilizado uma versão para avaliação.

Quadro 2 - Avaliação das plataformas NLP

Nome do NLP	Fornecedor	Critérios						
		1	2	3	4	5	6	7
Api.ai	Api.ai	sim	sim	sim	sim	sim	sim	sim
Cortana	Microsoft	não	não	sim	sim	não	não	sim
Google Now	Google Inc	não	não	sim	sim	sim	não	sim
Nuance Mix	Nuance Communications Inc	sim	N.A.	sim	sim	N.A.	N.A.	sim
Syn Engine	Synthetic Intelligence Network	sim	sim	sim	sim	sim*	não	não
Siri	Apple	não	não	sim	sim	não	não	sim

Fonte: Próprio autor. Legenda: N.A. Não Avaliado.

Já o Natural Language Processor Syn Engine, apesar de ser aderente a arquitetura do Assistente Financeiro Virtual, não é um serviço, ele é uma biblioteca de classes Java que não permite a ampliação do conhecimento de forma dinâmica, i.e. em tempo de execução, somente é possível com alteração no código de programação, em tempo de compilação.

Para este trabalho foi utilizado o serviço de Natural Language Processor da plataforma de assistentes virtuais, Api.ai, que é uma API disponibilizada através de *web services* que utilizam a arquitetura RestFul. Este serviço atendeu a todos os critérios da seleção.

Para utilizar o serviço do Natural Language Processor da Api.ai é necessário criar previamente uma conta de usuário desenvolvedor no website da própria Api.ai e obter chaves de acesso ao serviço, isto porque todas as perguntas submetidas ao NLP da Api.ai são autenticadas por estas chaves. O processo de autenticação da submissão da pergunta é realizado pelas bibliotecas disponibilizadas gratuitamente em diversas tecnologias pela Api.ai. Para que a submissão da pergunta ao NLP efetivamente aconteça é necessário configurar uma dessas bibliotecas com as chaves de autenticação.

Na época da codificação do Assistente Financeiro Virtual nenhuma das bibliotecas disponibilizadas pela Api.ai para acessar o seu serviço de processamento de linguagem natural eram compatíveis com Java, que foi a tecnologia utilizada para implementar o Coordinator e Dialog and Knowledge Manager. Para resolver esta restrição, decidiu-se construir um webservice, com tecnologia compatível com uma daquelas bibliotecas, com o objetivo de atuar como um proxy e autenticador dos serviços do Natural Language Processor para as classes Java do Assistente. Assim as chaves de autenticação foram registradas no código do webservice construído.

Dentre as bibliotecas disponibilizadas pela Api.ai optou-se em construir o *webservice* proxy / autenticador em linguagem Python pela maior familiaridade do autor com esta linguagem de programação. O *webservice* foi implementado compatível com o protocolo SOAP, que é um protocolo de transferência de mensagens em formato XML. O uso desse protocolo neste trabalho é justificado em razão do SOAP ser um padrão da indústria, muito utilizado em ambientes distribuídos.

O serviço da Api.ai define um assistente como um Agent, sendo que cada agent pode utilizar diversos domínios de conhecimento, como os domínios disponibilizados pela Api.ai, ou os domínios que podem ser criados. Cada agent pode processar textos somente em um único idioma e após ter sido criado não é possível alterar o idioma do agent. Para que o Assistente Financeiro Virtual reconheça as questões financeiras elaboradas nos idiomas inglês e português e contornar essa limitação do serviço, foram criados dois agents, um para a versão do domínio financeiro no idioma inglês, nomeado *AssistenteFinanceiroVirtual* e outro para a versão do domínio financeiro no idioma português, nomeado *FinancialVirtualAssistant*.

Para que o Natural Language Processor reconheça dados financeiros nas perguntas submetidas foi necessário construir previamente as versões do domínio de conhecimento financeiro. Quanto mais expressões típicas e termos do jargão financeiro estiverem representados na gramática e no dicionário, maior será a capacidade do Natural Language Processor em entender as perguntas financeiras do usuário.

Para ampliar o entendimento financeiro do Assistente Financeiro Virtual, alguns padrões de perguntas foram inseridos na sintaxe gramatical do NLP. Os padrões registrados na gramática da versão do agent em inglês seguiram a estrutura

da língua inglesa e para versão do agent em português os padrões registrados seguiram a estrutura de língua portuguesa.

Uma única versão do Assistente Financeiro Virtual acessa os dois agents configurados. Eles são diferenciados por códigos de identificação únicos, gerados pelo serviço do NLP da Api.ai, que são enviados em cada requisição ao *web service* do NLP, da Api.ai, de acordo com a linguagem escolhida pelo usuário.

O agent da versão em português, consegue reconhecer questões financeiras como: “Qual o lucro líquido da Petrobras em 2008?”, ou “Qual a dívida da Apple em 2009”. Para agent da versão em inglês reconhecer determinadas perguntas financeiras no idioma inglês, também foi necessária a criação de uma gramática e de um dicionário de termos específicos para o idioma, pois o Inglês possui estrutura gramatical diferente do idioma português. Os padrões inseridos na versão do idioma inglês, permitem que o NLP reconheça perguntas como: “What is the current assets of Apple in 2015”, ou “Show me the liabilities of Microsoft in 2016”.

Cada Natural Language Processor possui peculiaridades na construção da gramática e dos dicionários de termos compatíveis. A forma de codificação dos padrões de pergunta geralmente é específica para cada Natural Language Processor. Alguns utilizam expressões regulares, outros utilizam linguagens específicas para construção de gramáticas, como a JSpeech Grammar Format (JSGF) (W3C, 2000), ou definem regras próprias para criação de regras de sintaxe. O NLP da Api.ai permite a criação da gramática através do registro de padrões em um formato próprio, mas de fácil compreensão. A construção das gramáticas e dos dicionários foi realizada através de uma ferramenta web disponibilizada pelo própria Api.ai.

#### **4.3.2 Implementação do dicionário de termos**

Para que o Natural Language Processor reconheça as palavras de um domínio, ele irá recorrer para o dicionário de termos do domínio. O dicionário de termos é o local onde se registram os sinônimos dos conceitos do domínio, nomes, palavras-chaves e a definição de um valor padrão correspondente.

O NLP da Api.ai agrupou os termos em uma funcionalidade nomeada “Entity”, que corresponde a Entidade. Cada entity possui uma lista de termos, e cada termo possui um sinônimo ou uma lista de sinônimos. Quando o NLP reconhece um

sinônimo de um termo, ele o substitui pelo texto do valor padrão do termo, definido no dicionário.

Para o Assistente Financeiro Virtual definiu-se dois tipos de entidade: a entidade financeira, que é um tipo relacionado ao domínio financeiro e a entidade genérica, que pode ser utilizada em qualquer domínio. Os termos das entidades financeiras possuem como valor padrão um conjunto de informações estruturadas em formato texto, enquanto que os termos das entidades genéricas possuem como valores padrão, uma palavra ou um texto simples.

O objetivo da diferenciação é embutir no significado de cada termo da entidade financeira um conjunto de informações mínimas necessárias, que servirão como valores dos parâmetros obrigatórios nas consultas aos Financial Information Services implementados para o Assistente Financeiro Virtual. Ao utilizar o próprio Natural Language Processor para retornar as informações mínimas da empresa para a classe RequestParameters, apresentado na Seção 4.2.3 , evita-se a busca dessas informações e consegue-se diminuir a quantidade de requisições que os Financial Information Services fariam à camada Data. Esse menor acesso a camada Data contribui com a diminuição do tempo total que o Assistente Financeiro Virtual consome para responder à pergunta do usuário.

O conjunto de informações que correspondem ao valor padrão de cada entidade financeira pode ser embutida em um texto seguindo os mais diversos padrões estruturados disponíveis, e.g. XML, JSON. Para o Assistente Financeiro Virtual ela foi encapsulada em um texto no formato JSON. A justificativa pelo uso desse formato se deve ao fato do serviço do NLP da Api.ai já fornecer as respostas em JSON, o que torna mais fácil a integração das informações através da utilização do mesmo formato. Além disso, como o JSON utiliza menos caracteres para armazenar a mesma informação que o equivalente em XML, as trocas de informação, entre a classe Coordinator e o serviço NLP, tornam-se menos custosas e mais rápidas.

Foram configurados dois dicionários, um para o agent em inglês e outro por agent em português. As entidades configuradas nos dicionários não são iguais, são equivalentes, pois possuem diferenças relacionadas ao respectivo idioma. Para a versão do agent em inglês foram criadas as seguintes *entities* genéricas e seus respectivos significados:

1. CommandExpressions: Possui termos equivalentes a expressões de comando que podem ser elaboradas pelo usuário, e.g. “Show me”, “Inform”, “What is”. O Quadro 3 mostra os comandos registrados;
2. Greeting: São expressões opcionais que são utilizadas para iniciar um diálogo, e.g. Hello, Hi, Good morning.

Quadro 3 - Entity CommandExpressions

Expressão	Sinônimos	Significado
What is	What is, What's, Give me, Give, Would you tell me, Show me, Show, Inform, Say, Say me	Whatis

Fonte: Próprio autor.

As *entities* financeiras criadas na versão em inglês do dicionário foram: Company, USGAAP\_BalanceSheet, USGAAP\_IncomeStatements, Financial Ratios e YearPeriod.

A entity Company contém o nome e dados das empresas que são o alvo dos questionamentos dos usuários do Assistente Financeiro Virtual. Para facilitar a realização de testes e avaliações da implementação do Assistente, optou-se por incluir no dicionário, empresas cujo os respectivos dados estejam disponíveis na camada Data implementada. Para este trabalho, das empresas com dados disponíveis nessa camada, foram inseridas informações referentes a 30 empresas, 26 americanas e 4 brasileiras. A informação sobre quais as empresas a camada Data implementada disponibiliza os respectivos dados será apresentada na Seção 4.4.

O significado de cada termo da entity Company que foi inserido, informa o nome da empresa, que será apresentado na resposta final ao usuário, a Central Index Key (CIK), que é uma “chave”, ou o número de registro da empresa na U.S. Securities and Exchange Commission (US-SEC) (2016b), o código de identificação da ação da empresa e o código da bolsa de valores em que essa ação da empresa é negociada.

O código CIK foi incluído como informação mínima necessária, porque é um dos parâmetros obrigatórios para realizar consultas financeiras empresariais à camada Data implementada, que será apresentada na Seção 4.4. Também foi incluído o código da ação da empresa, o motivo é que ele é um dos identificadores

mais utilizados pelos sites de informações financeiras empresariais disponíveis no mercado. O Quadro 4 mostra um fragmento do dicionário da entity Company.

Quadro 4 - Fragmento da entity Company

Nome	Sinônimos	Significado
Apple	Apple, Apple Inc	{"name":"Apple Inc", "cik": "0000320193", "ticker":"aapl", "stockExchange":"NYSE"}
Bradesco	Bradesco, Banco Bradesco, Bank Bradesco	{"name":"Bank Bradesco", "cik": "0001160330", "ticker":"BBDO", "stockExchange":"NYSE"}
Gol AirLines	Gol Linhas Aéreas, Gol Linhas Aereas Gol, Gol AirLines	{"name":"Gol AirLines", "cik": "0001291733", "ticker":"gol", "stockExchange":"NYSE"}
PETROBRAS	BR, Petrobras, Petróleo Brasileiro, Petroleo Brasileiro SA,	{"name":"PETROBRAS - PETROLEO BRASILEIRO SA", "cik": "0001119639", "ticker":"pbr", "stockExchange":"NYSE"}
Vale S.A.	Vale S.A., Vale do Rio Doce, Vale	{"name":"Vale S.A.", "cik": "0000917851", "ticker":"VALE", "stockExchange": "NYSE"}

Fonte: Próprio autor.

A codificação para o significado de um termo na entity Company em formato JSON corresponde a:

```
{"name":"nome da empresa", "cik": "código CIK da  
empresa", "ticker":"código da ação", "stockExchange":"código da bolsa de  
valores"}
```

Por exemplo, para o Assistente Financeiro Virtual, o significado, ou valor padrão, da empresa Apple, corresponde a:

- name : "Apple Inc";
- cik : "0000320193";
- ticker : "aapl";

•stockExchange : "NYSE".

Esses dados foram registrados no dicionário da entity Company no formato JSON, conforme a seguir:

```
{ "name": "Apple Inc", "cik": "0000320193", "ticker": "aapl",
  "stockExchange": "NYSE" }
```

O Assistente Financeiro Virtual atualmente está apto a passar informações das nove empresas registradas nos dicionários, entretanto, para ampliar essa quantidade é necessário verificar primeiro se a camada Data do Assistente disponibiliza os dados da empresa desejada e posteriormente registrar os dados dela no formato padronizado no dicionário da entity Company.

A entity USGAAP\_BalanceSheet contém alguns conceitos financeiros utilizados nos balanços financeiros empresariais e a entity USGAAP\_IncomeStatements contém alguns conceitos financeiros utilizados nos demonstrativos financeiros empresariais. Esses conceitos seguem a taxonomia US-GAAP e a tecnologia XBRL.

Para que o dicionário contenha termos e palavras típicas do domínio financeiro é importante utilizar uma taxonomia conhecida e utilizada na área financeira. Para o Assistente Financeiro Virtual selecionou-se a taxonomia chamada US - Generally Accepted Accounting Practices (US-GAAP), que significa Práticas Contábeis Comumente Aceitas nos Estados Unidos. Segundo Geron et al. (2013, p 127-128), essa taxonomia foi criada pelo consórcio XBRL US (2016d) por contrato com a SEC dos Estados Unidos, mas, posteriormente, o Financial Accounting Standards Board (FASB) assumiu a responsabilidade pela taxonomia.

Qualquer outra taxonomia financeira pode ser utilizada no Assistente Financeiro Virtual, inclusive de forma simultânea, como a taxonomia das Normas Internacionais de Contabilidade, a International Financial Report Statement (IFRS) (2016a), emitidas pelo International Accounting Standards Board (IASB) (IFRS, 2016b). Entretanto, para que a taxonomia utilizada traga eficácia para as respostas do Assistente é necessário que ela seja suportada por uma tecnologia computacional padronizada que permita que as demonstrações financeiras das empresas sejam disponibilizadas em algum banco de dados disponível para consulta, e.g. a tecnologia XBRL.

Segundo Geron et al. (2013), a taxonomia financeira US-GAAP é, atualmente, a mais utilizada nas demonstrações financeiras empresariais disponibilizadas em tecnologia XBRL no mercado. Para que a implementação do Assistente Financeiro Virtual tenha acesso a uma maior quantidade de informações econômicas empresariais disponíveis possível, a taxonomia US-GAAP foi utilizada.

Para as *entities* financeiras USGAAP\_BalanceSheet e USGAAP\_IncomeStatements, o significado de cada termo corresponde a um conjunto de informações que contém o nome do elemento XBRL correspondente e o label, ou rótulo, relacionado.

O nome do elemento XBRL, que segue a taxonomia US-GAAP, é uma identificação única de um determinado conceito financeiro. Por exemplo, o conceito financeiro “Stockholders' Equity” é referenciado pelo nome do elemento XBRL “stockholdersequity”.

O label, ou rótulo, é o texto referente ao conceito financeiro que aparece nos demonstrativos financeiros produzidos pela tecnologia XBRL. Ele também será utilizado como este propósito nas respostas do Assistente Financeiro Virtual para o usuário.

Por exemplo, o conjunto de informações relativas ao termo “Long Term Debt”, é representado da seguinte forma:

- name : "LongTermDebtNoncurrent";
- label : "Long Term Debt".

Para o Assistente Financeiro Virtual, o dicionário da entity USGAAP\_BalanceSheet foi preenchido com 20 conceitos financeiros. O Quadro 5 mostra um fragmento dessa entity.

Quadro 5 - Fragmento da entity USGAAP\_BalanceSheet

<b>Termo</b>	<b>Sinônimos</b>	<b>Significado</b>
Assets	Assets, Total Assets, Assets Total	{"name":"assets","label":"Assets"}
Current Assets	Current Assets, Assets Current, Total Current Assets, Current Assets Total, AssetsCurrent	{"name":"assetscurrent","label":"Current Assets"}
Liabilities	Liabilities	{"name":"liabilities","label":"Liabilities"}

Long Term Debt	Long Term Debt, long-term debt	{"name":"LongTermDebtNoncurrent","label":"Long Term Debt"}
Stockholders' Equity	Stockholders' Equity, StockholdersEquity, Stockholders Equity, Equity, Total Equity, Equity Total	{"name":"stockholdersequity","label":"Stockholders' Equity"}

Fonte: Próprio autor.

Para a versão em português todos os termos e sinônimos das *entities* em inglês foram traduzidos para o idioma português. No caso das *entities* financeiras, o nome do elemento XBRL é o mesmo para as duas versões, entretanto, o label foi utilizado para mostrar a tradução do conceito financeiro. Por exemplo, na entity USGAAP\_BalanceSheet o termo “Current Assets” corresponde ao elemento XBRL “assetscurrent” e ao label “Current Assets”, conforme apresentado no Quadro 5. Na versão traduzida dessa entity, nomeada USGAAP\_BalancoFinanceiro, o termo “Ativo Circulante”, que é a tradução do termo “Current Assets”, corresponde ao elemento XBRL “assetscurrent” e ao label “Ativo Circulante”, conforme apresentado no Quadro 6.

Quadro 6 - Fragmento da entity USGAAP\_BalancoFinanceiro

<b>Termo</b>	<b>Sinônimos</b>	<b>Significado</b>
Ativo	Ativo, Total de Ativos, Ativos Total	{"name":"assets","label":"Ativo"}
Ativo Circulante	Ativo Circulante, Total do ativo circulante, Total ativo circulante, ativo circulante total	{"name":"assetscurrent","label":"Ativo Circulante"}
Passivo	Passivo	{"name":"liabilities","label":"Passivo"}

Fonte: Próprio autor.

Outras *entities* foram criadas, como o Financial Ratios, que é uma lista de alguns indicadores financeiros para avaliação da situação econômica das organizações.

Foi criada também a entity YearPeriod que contém os termos e os respectivos símbolos dos períodos do ano comumente utilizados em finanças como “trimestre”, “semestre”. Alguns itens da entity YearPeriod estão listados no Quadro 7.

Quadro 7 - Parte da entity YearPeriod

Período do ano	Sinônimos	Significado
The first semester	S1, The first semester, first semester	S1
The second semester	S2, The second semester, second semester	S2
The first quarter	Q1, the first quarter, first quarter	Q1
The second quarter	Q2, the second quarter, second quarter	Q2
The fourth quarter	Q4, The fourth quarter, fourth quarter	Q4

Fonte: Próprio autor.

### 4.3.3 Regras gramaticais

Após o processo de identificação das *entities*, o Natural Language Processor compara a pergunta do usuário com os padrões de perguntas registrados na gramática. Por se tratar de um Assistente Financeiro Virtual, os padrões foram elaborados para que o jargão financeiro fosse reconhecido.

Para a versão da gramática em português, a pergunta “Qual o Passivo da Petrobras em 2015?” é capturada pelo NLP através da seguinte regra configurada:

```
[@ExpressaoDeComando] [o,a] @USGAAP_BalancoFinanceiro [para]
[pra,pro,o,a,da,de,do,o do,o da] @Empresa:Empresa [pra,pro,o,a,do,o do,no]
[@periodo:periodo] [do,da,de,em] [@sys.number:number]
```

Para a pergunta em inglês, “What is the current asset of Apple in 2016?”, a seguinte regra consegue reconhecê-la:

```
[@greeting] [@CommandExpressions]
@USGAAP_BalanceSheet:financialConcept [of] @Company:companyData
[company] [in,on] @YearPeriod:yearPeriod [at,in,on fiscal year, in fiscal
year,of fiscal year,of year, year] @sys.number:year
```

A posição das palavras e dos nomes das *entities*, além da nomeação dos parâmetros aguardados definem um padrão de perguntas esperado. Um símbolo “@” seguido por um nome identifica uma *entity* que é esperada naquela posição. E o

símbolo “:” seguido por um nome identifica um parâmetro que será preenchido com o significado do termo da entity reconhecido. Todas as palavras entre os símbolos de colchete, “[ ]” sinaliza para o NLP que são opcionais. As palavras entre vírgulas dentro de colchetes sinalizam que opcionalmente qualquer uma delas pode aparecer naquela posição. Palavras entre vírgulas dentro de símbolos chaves “{ }” significa que naquela posição é obrigatório o aparecimento de pelo menos uma das palavras daquele conjunto.

#### 4.3.4 Reconhecimento das intenções do usuário

O Natural Language Processor, da Api.ai, agrupa regras em um conceito chamado “Intent”, que significa intenção, que permite associar um conjunto de padrões de perguntas a uma Intent. Quando o NLP reconhece uma pergunta do usuário é possível também supor qual a intenção do usuário. Além disso, é um recurso útil para identificar o que cada conjunto de padrão consegue capturar. O nome de uma Intent pode conter espaços e caracteres acentuados da língua portuguesa. No Quadro 8 são apresentadas algumas regras da intent “What the Financial Concept of Balance Sheets Company At Period”.

Quadro 8 - Parte da Intent “What is the Financial Concept of Company At Period”

Id	Regra
1	[ @greeting ] [ @CommandExpressions ] @USGAAP_BalanceSheet:financialConcept [ of ] @Company:companyData [ company ][ in,on ] @YearPeriod:yearPeriod [ at,in,on fiscal year, in fiscal year,of fiscal year,of year, year ] @sys.number:year
2	[ @greeting ] [ @CommandExpressions ] @USGAAP_BalanceSheet:financialConcept [ of ] @Company:companyData [ company ] [ at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year ] @sys.number:year
3	[ @greeting ] [ @CommandExpressions ] @Company:companyData [ company ] @USGAAP_BalanceSheet:financialConcept [ at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year ] @sys.number:year
4	[ @greeting ] [ @CommandExpressions ] @Company:companyData [ company ] @USGAAP_BalanceSheet:financialConcept [ in,on ] @YearPeriod:yearPeriod [ at,in,of,on fiscal year, in fiscal year,of fiscal year,of year, year ] @sys.number:year

Fonte: Próprio autor.

A fim de evitar a duplicação de trechos de código por idioma, e dificultar a manutenção, as classes implementadas trabalham com a mesma informação

provinda do Natural Language Processor, independente do idioma. Para contribuir com essa visão, padronizou-se os nomes dos parâmetros que são encaminhados para a camada Orchestration nos dois agents do NLP. Os nomes dos parâmetros são: `financialConcept`, `companyData` e `year`. Conforme os Financial Information Services implementados, que serão apresentados na Seção 4.3.5, buscam esses parâmetros e seus respectivos valores para executar suas consultas à camada Data.

Independente do agent, em inglês ou português, do NLP que irá processar a pergunta do usuário, todos os termos reconhecidos são transformados em valores dos parâmetros especificados acima. Por exemplo, se a expressão do usuário “Qual o Passivo da Petrobras em 2015?” for processada pelo agent, da versão em português, do Natural Language Processor, ele irá identificar que a expressão está mapeada por uma das regras da intent “Qual é o item contábil da empresa no período” e extrairá os parâmetros esperados, conforme a seguir:

```
$financialConcept = {"name": "liabilities", "label": "Passivo"}
$companyData = {"name": "PETROBRAS - PETROLEO BRASILEIRO SA",
"cik": "0001119639", "ticker": "pbr", "stockExchange": "NYSE"}
$year = 2015
```

Se a expressão do usuário “Show me the assets of Vale in 2009” for processada pelo agent, da versão em inglês, do NLP, este irá reconhecer que a expressão corresponde a uma das regras da intent “What is the Financial Concept of Company At Period” e irá extrair os parâmetros esperados. Os valores dos parâmetros corresponderão a:

```
$financialConcept = {"name": "assets", "label": "Ativo"}
$companyData = {"name": "Vale S.A.", "cik": "0000917851", "ticker": "VALE",
"stockExchange": "NYSE"}
$year = 2009
```

Para este trabalho foram criadas 28 (vinte e oito) intents, sendo 14 (quatorze) para cada versão do idioma, conforme apresentadas no Quadro 9 e no Quadro 10.

Quadro 9 - Lista das Intents da versão em inglês

<b>Id</b>	<b>Nome da Intent</b>
1	What is the Financial Concept of Company At Period
2	What is the Variation of Financial Ratio of Company At last Periods
3	What is the Variation of Financial Ratio of Company At last Period
4	Change Company
5	Change Company And Financial Concept
6	Change Company And Period
7	Change Financial Concept
8	Change Financial Ratio
9	Change The Financial Concept And The Period
10	Change The last period
11	Change The last periods
12	Change The period
13	Inform About Company
14	Inform About Financial Concept

Fonte: Próprio autor.

Quadro 10 - Lista das Intents da versão em português

<b>Id</b>	<b>Nome da Intent</b>
1	Qual é o item contábil da empresa no período
2	Qual é a variação do indicador financeiro da empresa nos últimos períodos
3	Qual é a variação do indicador financeiro da empresa no ultimo período
4	Altere a empresa
5	Altere a empresa e o item contábil

6	Altere a empresa e o período
7	Altere o Indicador Financeiro
8	Altere o item contábil
9	Altere o item contábil e o período
10	Altere o período
11	Altere o ultimo período
12	Altere os últimos períodos
13	Informe Dados da Empresa
14	Informe Dados sobre Conceito Financeiro

Fonte: Próprio autor.

As intents iniciadas com as palavras “What is”, na versão em inglês, e com as palavras “Qual é”, na versão em português, são intents com padrões que capturam perguntas completas, isto é, perguntas em que o usuário informa todos os dados suficientes para que o Assistente Financeiro Virtual consiga trazer a resposta. Normalmente são as perguntas que iniciam os questionamentos do usuário, por exemplo, “Qual o passivo da Gerdau no segundo trimestre de 2014?”. Nessa pergunta o usuário informa o conceito contábil, a empresa e o período específico, isto é, todos os parâmetros necessários para que a consulta seja realizada com sucesso.

Entretanto, em uma conversa natural, as perguntas subsequentes podem ser incompletas, isto é, o interlocutor pode, por exemplo, perguntar pelo passivo da Gerdau em outro período simplesmente perguntando assim, “e em 2013?”. Esta é uma situação comum em um diálogo. E o comportamento esperado de um assistente é que ele mantenha o diálogo com o usuário, isto é, é esperado que o Assistente tenha a capacidade de responder perguntas em que o usuário omitiu informações já passadas por ele em perguntas anteriores. Neste caso, através dos padrões da intent “Altere o período”, o Natural Language Processor irá entender que o usuário quer apenas alterar o período.

As intents iniciadas com a palavra “Change”, na versão em inglês, e com a palavra “Altere”, na versão em português, são intents com padrões que capturam perguntas que tem a intenção de alterar algum parâmetro da pergunta anterior.

Um recurso disponibilizado pelo NLP da Api.ai é um parâmetro associado a cada intent nomeado de “action”. Ele é um parâmetro de configuração único de cada intent, cujo o valor é definido pelo configurador da gramática. O valor dele não é influenciado pelas perguntas submetidas ao NLP, assim como o valor do nome da intent. Diferente da variável que armazena o nome da intent, o parâmetro action não pode ter espaços ou caracteres especiais. Para o Assistente Financeiro Virtual, este recurso foi utilizado para identificar a intenção do usuário, de forma única e independente do idioma. Para facilitar o sincronismo dentro da mesma camada, o valor de cada intent, correspondente a um conjunto de perguntas completas, foi configurado com o nome da classe que implementa o Financial Information Service que irá construir a resposta dessas perguntas. Os Financial Information Services serão apresentados na Seção 4.3.5 .

Por exemplo, a intent “Qual é o item contábil da empresa no período”, da versão português, e a intent “What is the Financial Concept of Company At Period”, da versão em inglês, possuem o mesmo valor “CompanyRatioInformationService” para o parâmetro action, que corresponde ao nome da respectiva classe Java que irá construir a resposta dessas perguntas.

Para este trabalho, na configuração da gramática foi feita uma associação entre os nomes das intenções e os valores dos respectivos parâmetros action, conforme apresentado no Quadro 11.

Quadro 11 - Nome da intent e o respectivo valor do parâmetro action

<b>Nome da intent</b>	<b>Valor do parâmetro action</b>
What is the Financial Concept of Company At Period	CompanyRatioInformationService
Qual é o item contábil da empresa no período	CompanyRatioInformationService
Inform About Company	CompanyInformationService
Informe Dados da Empresa	CompanyInformationService
Inform About Financial Concept	FinancialConceptInformationService

Informe Dados sobre Conceito Financeiro	FinancialConceptInformationService
---	------------------------------------

Fonte: próprio autor.

Exemplo de uma lista de parâmetros enviada pelo Natural Language Processor ao Coordinator em resposta à pergunta “Qual o passivo do twitter em 2015?” submetida pelo usuário:

- intent: Qual é o item contábil da empresa no período;
- action: CompanyRatioInformationService;
- companyData: { "name": "TWITTER, INC.", "cik": "0001418091", "ticker": "TWTR", "stockExchange": "NYSE"};
- financialConcept: {"name":"liabilities","label":"Passivo"};
- year: 2015.

Após encaminhar esses parâmetros para o Coordinator a participação do NLP nessa transação termina.

#### **4.3.5 Implementação do *Financial Information Services***

Os responsáveis por realizar ou fornecer resultados de análises financeiras são os Financial Information Services, e para que as análises mais complexas de risco, realizadas sob a ótica de muitas dimensões, ou visões de negócio, típicas de avaliações financeiras, sejam feitas com flexibilidade e performance aceitáveis, as consultas OLAP (Online Analytical Processing) são as mais adequadas. Entretanto, como as fontes de dados utilizadas nesse Assistente são os artefatos ou repositórios em XBRL, documentos baseado em XML, então realizar consultas OLAP nestes dados não é algo trivial.

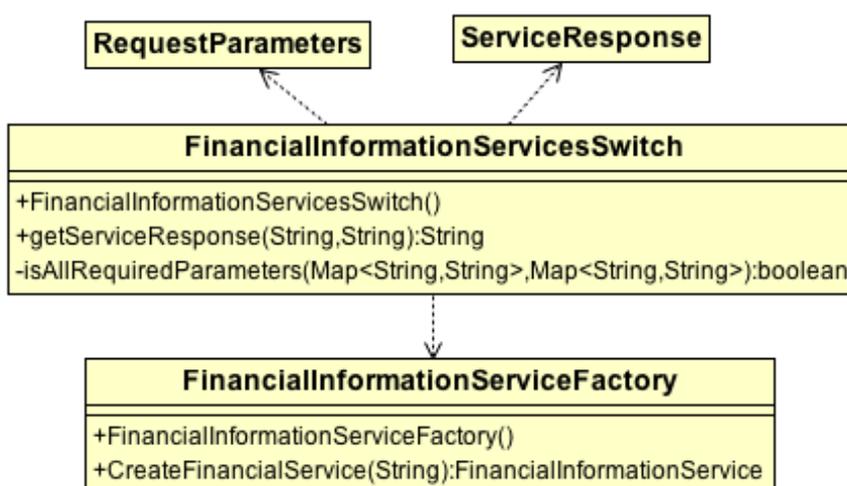
Uma solução para esta questão é a LMDQL, uma linguagem especificada por (SILVA, 2010) que permite realizar consultas analíticas OLAP em documentos XBRL. Outra grande contribuição desse trabalho foi definição de operadores específicos para área de análise financeira, diminuindo consideravelmente a complexidade das queries (SILVA, 2010) definiu operadores que permitem ao assistente responder questões financeiras, como o operador GrauAlavancagemFinanceira que calcula o nível de endividamento de uma empresa. (SILVA, 2010) definiu também operadores mais complexos como os de modelos

estatísticos de avaliação de empresas que permitem ao assistente a "capacidade" de indicar empresas com boa saúde financeira, ampliando a sofisticação da consultoria financeira.

Assim, para que o Assistente forneça esse tipo de resposta, é necessário que o algoritmo de construção das consultas OLAP, utilizando os operadores em LMDQL citados, em uma base XBRL esteja implementado nas classes de serviço de informação financeira. A implementação dessas classes com o algoritmo LMDQL/OLAP está prevista como um trabalho futuro.

Para representar o conjunto dos Financial Information Services foi implementado um *web service*, compatível com o protocolo SOAP, através da classe `FinancialInformationServicesSwitch`, apresentada no diagrama de classes ilustrado na Figura 28.

Figura 28 - Diagrama de classes relacionadas com a classe `FinancialInformationServicesSwitch`



Fonte: Próprio autor.

A função do *web service* é retornar uma resposta de um Financial Information Service, mediante os parâmetros de entrada do serviço.

Esse *web service* possui um método nomeado `getServiceResponse` que exige dois parâmetros de entrada, conforme ilustrado na Figura 28, que são duas *strings*: a primeira *string*, nomeada `serviceName`, corresponde ao nome do serviço e a segunda *string*, nomeada `requestParameters`, corresponde a uma instância da classe `RequestParameters`, apresentada na Seção 4.2.3, que deve ser encapsulada no formato JSON. A resposta desse *web service* corresponde a uma instância da

classe `ServiceResponse`, apresentada na Seção 4.2.7 , que é disponibilizada encapsulada em formato JSON.

Para retorna o valor de um `Financial Information Service`, a classe `FinancialInformationServicesSwitch` seleciona e instancia uma das classes do tipo `FinancialInformationService`, que será apresentada na Seção 4.3.6 , que é uma classe que implementa um `Financial Information Service`, e, em seguida, encaminha os parâmetros da consulta para a instância recém criada. Ao receber o resultado, a classe `FinancialInformationServicesSwitch` o encaminha como resposta do *web service*.

Para implementar esta comutação de classes do tipo `FinancialInformationService` mediante o fornecimento de uma *string* como parâmetro de entrada, criou-se uma classe nomeada `FinancialInformationServiceFactory`, que segue um *design pattern* do tipo *Factory Pattern*.

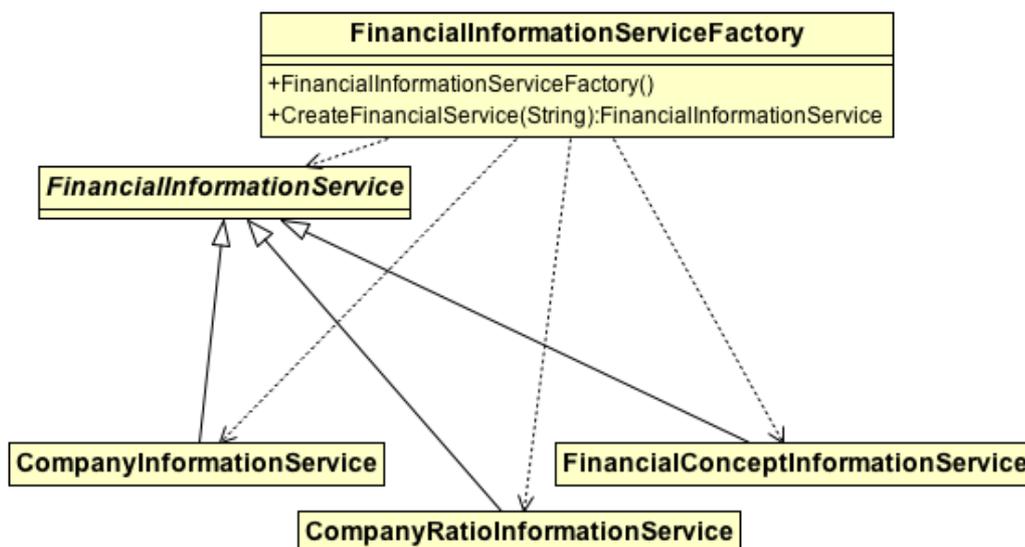
Conforme Lasater (2007), *design patterns* são modelos lógicos de implementação cujo o objetivo é facilitar a construção e a manutenção do código. Eles padronizam as tarefas comuns de programação em formas reconhecíveis dando maior coesão aos projetos de software. O *Factory Pattern*, segundo Lasater (2007), é uma classe que constrói e retorna uma instância de um tipo de classe. A vantagem de utilizá-lo, conforme Lasater (2007) é consolidar a lógica da implementação em um único lugar, na fábrica, obtendo assim a uniformidade e a manutenibilidade do código.

Em razão do mapeamento dos serviços de informações financeiras ser realizado por uma classe padrão tipo fábrica, para adicionar novos comportamentos para o Assistente Financeiro Virtual é necessário registrar nesta fábrica a nova classe que representa o `Financial Information Service` a ser incluído, i.e. todos os `Financial Information Services` que o Assistente Financeiro Virtual pode acionar devem estar registrados na classe `FinancialInformationServiceFactory`.

A classe fábrica `FinancialInformationServiceFactory` implementada tem a função de retornar uma instância de uma classe, do tipo `FinancialInformationService`, mediante o recebimento do nome da classe como parâmetro. Este nome, está associado a uma intenção do usuário e é informado pelo `Natural Language Processor`. Essa associação foi apresentada na Seção 3.3.3 .

A Figura 29 ilustra a fábrica de serviços de informação financeira e as subclasses, da classe `FinancialInformationService`, que podem ser acionadas. Essas subclasses serão apresentadas na Seção 4.3.6

Figura 29 - Diagrama de classes da fábrica de serviços financeiros



Fonte: Próprio autor.

Seguindo o fluxo de informação, a partir de quando a classe `FinancialInformationServicesSwitch`, que é um *web service*, é invocada pelo método `getServiceResponse`, ela recebe o nome da classe relacionada a intenção identificada atual, e encaminha para a classe fábrica `FinancialInformationServiceFactory`. Caso a fábrica não encontre nos seus registros o nome da classe que representa o `Financial Information Service` informado, ocorrerá uma situação de exceção, isto é, uma falha de mapeamento. Esta exceção disparará um objeto da classe `UnknownFinancialInformationServiceException`, que foi criada especificamente para tratar este erro. Nesse caso, como tratamento da ocorrência dessa exceção, a classe `FinancialInformationServicesSwitch` retornará um objeto `ServiceResponse` para o cliente do *web service* com uma mensagem informando que o Assistente conseguiu identificar a pergunta enviada, mas que ainda não conhece uma resposta para ela.

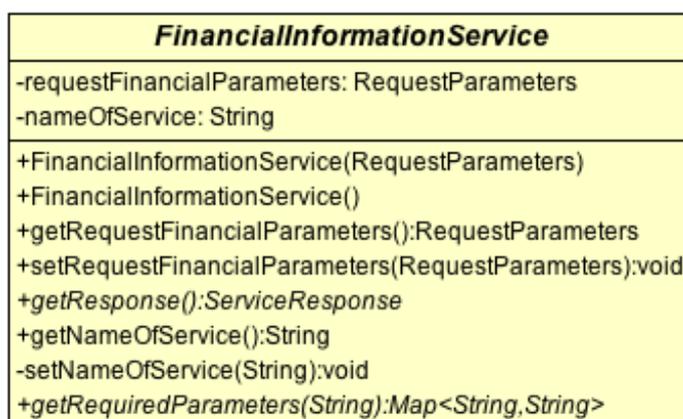
Caso o nome da classe seja encontrado pela fábrica, esta retorna uma instância da respectiva classe `FinancialInformationService` para a `FinancialInformationServicesSwitch`, que encaminha para esta última os parâmetros da requisição financeira, embutidos em um objeto da classe `ParametersRequest`, e aguarda a resposta. Após receber como resposta um objeto `ServiceResponse` a

classe `FinancialInformationServicesSwitch` o converte em um texto no formato JSON e o retorna como resposta para o cliente do *web service*.

#### 4.3.6 Classe *FinancialInformationService*

A classe `FinancialInformationService` representa um Financial Information Service, e todas as classes que a implementam são especializadas no domínio financeiro do Assistente Financeiro Virtual. A classe `FinancialInformationService`, conforme ilustrada na Figura 30, é uma classe abstrata (e por isso não pode ser instanciada) que implementa alguns métodos construtores e métodos para manipular parâmetros da consulta. Ela obriga que as suas subclasses implementem dois métodos abstratos: `getRequiredParameters`, para informar a lista de parâmetros obrigatórios, e `getResponse`, para fornecer um objeto de resposta.

Figura 30 - Classe abstrata `FinancialInformationService`



Fonte: Próprio autor.

O método `getRequiredParameters` é utilizado para evitar que a classe seja instanciada sem que todos os parâmetros obrigatórios tenham sido informados. A verificação é realizada pela classe `FinancialInformationServicesSwitch` após receber da fábrica `FinancialInformationServiceFactory` uma instância da classe `FinancialInformationService`. A classe `FinancialInformationServicesSwitch` chama o método `getRequiredParameters` da classe `FinancialInformationService` para realizar a verificação.

As implementações das classes `FinancialInformationService` são responsáveis por elaborar as respostas que serão encaminhadas para o usuário, e, também, por fornecer respostas completas no idioma, selecionado pelo usuário. As classes `FinancialInformationService` dispõem da classe `RequestParameters` para

obter os parâmetros necessários para realizar as consultas aos serviços financeiros. Conforme apresentada na Seção 4.2.3, a classe `RequestParameters` fornece, através do método `getParameters`, uma estrutura de dados do tipo tabela *Hash*, onde o nome de cada parâmetro é a chave para obtenção do seu respectivo valor. A classe `RequestParameters` também disponibiliza a informação relacionada ao idioma do usuário, através do método `getLanguage`.

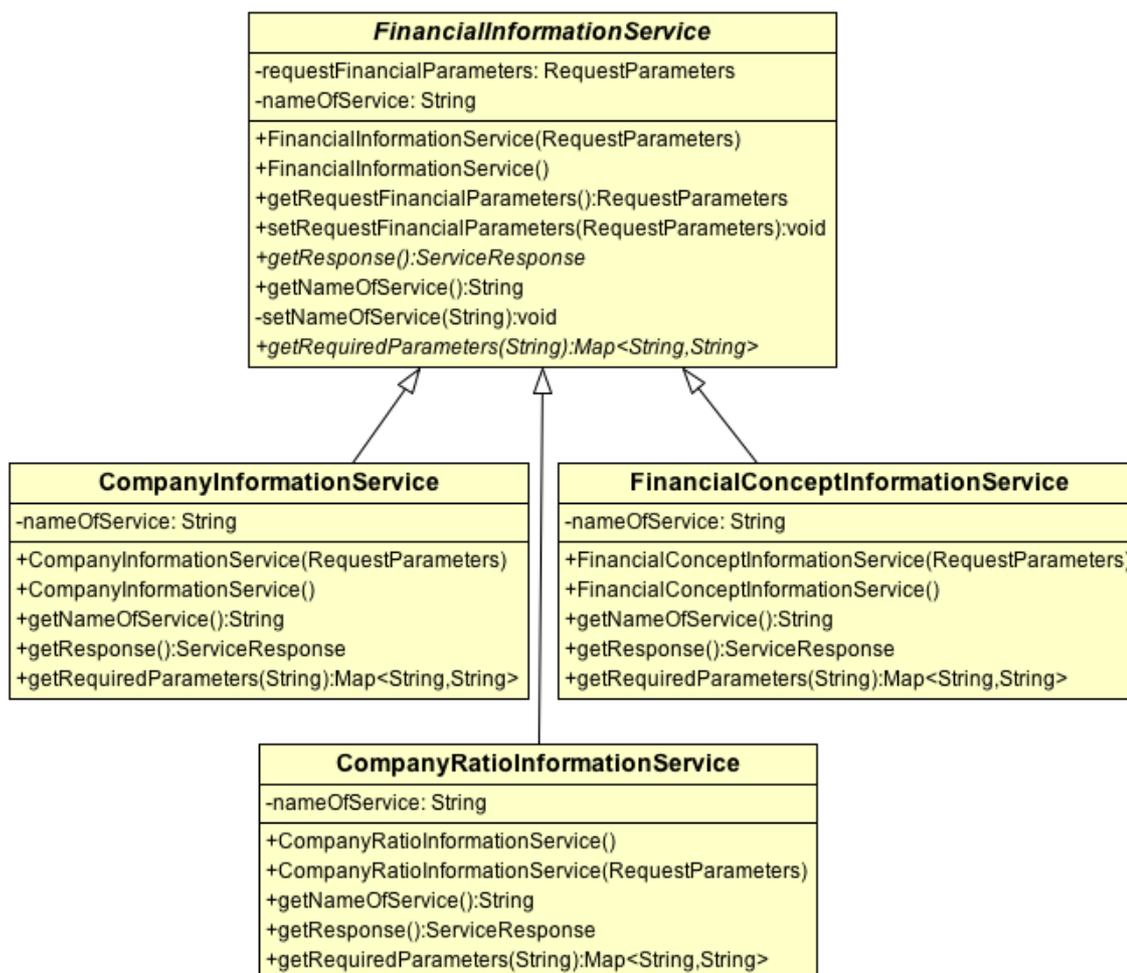
Para que o Assistente funcione integrado, os mesmos nomes dos parâmetros buscados pelas implementações da classe `FinancialInformationService` devem ser iguais aos nomes dados aos parâmetros inseridos na gramática do processador de linguagem natural, apresentada na Seção 4.3.3. Todos os nomes dos parâmetros esperados por um `FinancialInformationService` devem estar configurados nessa gramática.

Para este trabalho foram implementadas três subclasses da classe `FinancialInformationService`, conforme o diagrama de classes ilustrado na Figura 31. A classe `CompanyRatioInformationService`, que traz como resposta informações relacionadas ao índice, ou conceito, financeiro de uma empresa em determinado período, a classe `CompanyInformationService`, que informa dados sobre a empresa consultada, e a classe `FinancialConceptInformationService`, que fornece a definição sobre o conceito financeiro consultado.

Para obter os dados que irão compor as respostas, todas essas subclasses implementadas trocam informações com os *web services* implementados para representar os serviços de informação financeira, da camada *Understanding / Knowledge*, através do protocolo SOAP.

Como o objetivo de esclarecer como uma classe do tipo `FinancialInformationService` obtém os parâmetros da consulta e acessa os serviços de informação financeira, a classe `CompanyRatioInformationService` será apresentada. Como elas funcionam de forma semelhante, somente se diferenciando em como cada uma trata os dados e constrói as respostas, as classes `CompanyInformationService` e `FinancialConceptInformationService` não serão apresentadas.

Figura 31 - Diagrama das classes Financial Service implementadas



Fonte: Próprio autor.

A classe `CompanyRatiInformationService` disponibiliza respostas com texto em linguagem natural que contém o valor do índice, ou conceito, financeiro de uma empresa em um período especificado. A classe consegue construir as mensagens das respostas do Assistente nos idiomas português e inglês.

A classe `CompanyRatiInformationService`, após ser chamada pelo método `getResponse`, busca na tabela *Hash*, obtida da classe `RequestParameters`, as chaves `companyData`, `financialConcept` e `year` e obtém os respectivos valores.

Os valores dos parâmetros `companyData` e `financialConcept`, conforme apresentados nas Seções 4.3.2 e 4.3.3, que tratam sobre o dicionário de termos e a gramática do processador de linguagem natural, não são textos simples, eles são textos em formato estruturado, que seguem a notação JSON, que encapsulam informações consolidadas a respeito de uma entidade. No contexto do Assistente Financeiro Virtual, entende-se por entidade os itens fundamentais para o domínio

financeiro, os quais correspondem a Empresa (Company) e Conceito Financeiro (FinancialConcept), conforme apresentada na Seção 3.3.2 .

Os dados estruturados nos parâmetros `companyData` e `financialConcept`, correspondem as respectivas instâncias das classes `CompanyData`, que representa a entidade `Company`, e `FinancialConcept` que representa a entidade `Financial Concept`.

Tomando como exemplo o seguinte texto extraído do parâmetro `companyData` de uma entidade `Company`:

```
companyData: { "name": "TWITTER, INC.", "cik": "0001418091", "ticker":  
"TWTR", "stockExchange": "NYSE" }
```

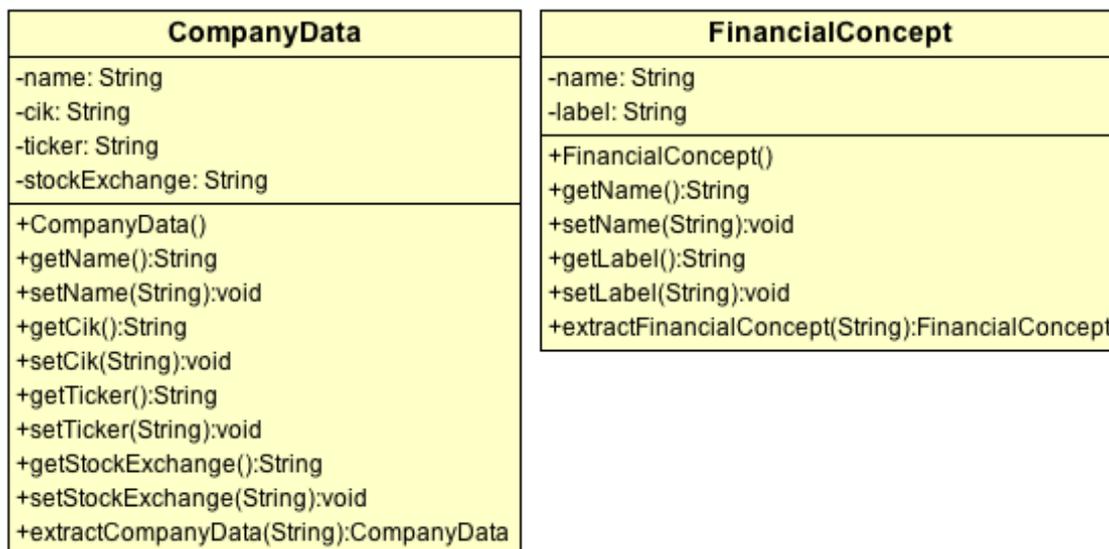
Após a conversão do texto em classe Java, a instância da classe `CompanyData` resultante terá as suas variáveis privadas com os seguintes valores atribuídos:

- `name`: "TWITTER, INC.";
- `cik` = "0001418091";
- `ticker` = "TWTR";
- `stockExchange` = "NYSE".

As classes `CompanyData` e `FinancialConcept`, ilustradas na Figura 32, foram implementadas e possuem, cada uma, um método que retorna uma instância da sua respectiva classe mediante a submissão de um texto, em formato JSON, compatível com a sua estrutura. O método `extractCompanyData`, para a classe `CompanyData` e o método `extractFinancialConcept`, para a classe `FinancialConcept` fazem a conversão de um texto em uma classe Java da mesma forma como é feita na classe `NLPServiceAdapter`, apresentada na Seção 4.2.4 .

Com o algoritmo de conversão dentro das próprias classes das entidades retira-se das classes `FinancialInformationService`, a responsabilidade e a especificidade em realizar esse processo, e amplia-se a manutenibilidade em razão do código estar em um único lugar.

Figura 32 - Classe CompanyData e FinancialConcept



Fonte: Próprio autor.

De posse das instâncias das classes representantes da empresa, do conceito financeiro e do período, a classe `CompanyRatioInformationService` extrai as informações necessárias para submeter uma consulta ao *web service* `XbrlUsgaapWebService`, da camada `Data`. Esse *web service* foi implementado e será apresentado na Seção 4.4. Ele possui um método chamado `getCompanyFinancialData` que exige os parâmetros `USGAAPElementName`, `cik`, `year` e `period` para retornar o resultado.

A classe `CompanyRatioInformationService` configura o valor do parâmetro `USGAAPElementName` como o valor da resposta do método `getName` da classe `FinancialConcept`, o valor do parâmetro `cik` é configurado com o valor retornado do método `getCik` da classe `CompanyData`, e os valores dos parâmetros `year` e `period` são os mesmos obtidos da classe `RequestParameters`.

A resposta do *web service* é um arquivo XML contendo informações financeiras detalhadas sobre o evento financeiro pesquisado e somente no idioma inglês, conforme exemplo reproduzido no Anexo A. Com as informações desse arquivo, e as informações das classes das entidades, a classe `CompanyRatioInformationService` constrói a resposta. Durante essa construção, em razão do Assistente ser multilíngue, a classe `CompanyRatioInformationService` utiliza o método `getLabel` da classe `FinancialConcept` para inserir na resposta o nome do conceito financeiro correspondente ao idioma do usuário.

O seguinte exemplo é um texto de resposta retornado pela classe `CompanyRatioInformationService`:

*The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00.*

Esse texto foi construído pela classe `CompanyRatioInformationService` em respostas aos seguintes parâmetros e seus respectivos valores obtidos da classe `RequestParameters`:

- `companyData = {"name": "PETROBRAS - PETROLEO BRASILEIRO SA", "cik": "0001119639", "ticker": "pbr", "stockExchange": "NYSE"};`
- `financialConcept = {"name": "assetscurrent", "label": "Current Assets"};`
- `year = 2009.`

#### 4.4 IMPLEMENTAÇÃO DA CAMADA DATA

Para atender as especificações do Assistente Financeiro Virtual, conforme apresentadas na Seção 1.2, os XBRL Repositories, da camada Data, devem ser disponibilizados e administrados por instituições idôneas e certificadas, e, preferencialmente, periodicamente atualizados por elas, o que garante que as respostas são baseadas em dados atuais e autênticos.

Para esta camada são previstos serviços ou bancos de dados que disponibilizem dados em XBRL através de consultas realizadas por parâmetros ou por queries de banco de dados. Esta camada pode ser implementada através de um banco de dados em XBRL que retorna os dados solicitados mediante uma query, compatível, construída com os parâmetros enviados pela camada Understanding / Knowledge. Entretanto, para este trabalho, para facilitar o acesso a dados em XBRL, disponibilizados conforme as especificações do Assistente, foi utilizado como fonte de dados do Assistente Financeiro Virtual um *web service* disponível no mercado, i.e., a base de dados utilizada pelo Assistente é externa.

##### **4.4.1 Classe *XbriUsgaapWebService***

Para representar a camada Data, foi implementada a classe `XbriUsgaapWebService`, ilustrada na Figura 33, que é um *web service* que

disponibiliza dados da US-SEC através do protocolo SOAP. Para acessar esses dados o *web service* possui o método `getCompanyFinancialData` que aceita os seguintes parâmetros de entrada: `USGAAPElementName`, que corresponde ao nome do elemento XBRL compatível com a taxonomia US-GAAP; `cik`, que é o código CIK da empresa, `year`, que corresponde ao ano da consulta; `period`, que corresponde ao período do ano. O resultado desse *web service* é um texto em XML com dados financeiros estruturados.

Figura 33 - Classe `XbrlUsgaapWebService`

<b>XbrlUsgaapWebService</b>
-XbrlUsApiKey: String
-XbrlUsApiUrl: String
-HTTP_SUCESS_COD: int
+XbrlUsgaapWebService()
+getCompanyFinancialData(String,String,String,String):String

Fonte: Próprio autor.

Para disponibilizar os dados, a classe `XbrlUsgaapWebService` acessa outro *web service*, o *web service* do XBRL-US (2016b) chamado XBRL US API, que torna disponível um banco de dados composto por todos os demonstrativos financeiros de empresas em formato estruturado XBRL. Este serviço, chamado Data Analysis Toolkit (XBRL-US, 2016b) fornece dados de relatórios financeiros corporativos em XBRL públicos.

Conforme o XBRL-US, a SEC obriga as empresas que emitem e/ou transacionam ações no mercado americano a enviar para ela as respectivas demonstrações financeiras com o objetivo de analisar a sua conformidade, bem como para identificar anomalias e distorções nos valores que podem indicar uma fraude contábil.

O motivo da seleção do serviço XBRL US API foi em razão: da origem dos dados corresponderem a uma instituição de controle financeiro confiável; da facilidade de acesso aos dados dos demonstrativos financeiros empresariais; da utilização de um padrão estruturado de dados, o XBRL; da frequente atualização do banco de dados do serviço que é realizada a cada quinze minutos. As características desse serviço permitem atender as especificações do Assistente Financeiro Virtual, conforme apresentadas na Seção 1.2.

O acesso da classe `XbrlUsgaapWebService` ao *web service* da XBRL-US é realizado através de uma API, disponibilizada em tecnologia RestFul, onde o Uniform Resource Locator (URL) corresponde a uma query de consulta. Para a classe `XbrlUsgaapWebService` realizar a consulta ao *web service*, ela monta um URL composto pelo endereço do servidor, pelos dados de autenticação de um usuário pré-cadastrado do serviço da XBRL-US e, principalmente, pelos parâmetros da consulta.

O resultado do serviço da XBRL-US é fornecido em formato XML. Este documento XML é encaminhado intacto como resposta da classe `XbrlUsgaapWebService`.

Algumas exceções, geradas pelo serviço da XBRL-US, podem ocorrer e.g. não haver dados disponíveis sobre determinada empresa no período desejado, a exemplo de empresas brasileiras que possuem identificador (CIK), mas que só disponibilizaram informações financeiras para SEC numa fase experimental entre 2008 e 2009. Para esses casos, o arquivo XML é enviado vazio.

Tratar essas exceções é uma das funções da classe `XbrlUsgaapWebService`, que sinaliza para o cliente do *web service* a ausência de dados. Além disso, a classe `XbrlUsgaapWebService` é a responsável por inserir o código de autenticação de usuário na query de consulta ao serviço da XBRL-US.

#### 4.5 FUNCIONAMENTO INTEGRADO DOS COMPONENTES E SERVIÇOS

Para facilitar o entendimento de como ocorre a integração entre os componentes e serviços implementados, será apresentado, a seguir, o exemplo de uma sequência de atividades executadas internamente no Assistente, em resposta a uma pergunta do usuário, conforme o diagrama de sequência ilustrado na Figura 34.

- 1.A sequência inicia quando o usuário submete a pergunta “Give me the Current Assets of Petrobras in 2009” na aplicação cliente, que é uma implementação do User Interface Manager, executada no navegador web. Esta aplicação encaminha o texto da pergunta para o servidor via protocolo Http.
- 2.Uma instância da classe `Coordinator`, que está em execução no servidor aguardando as requisições, recebe o texto da pergunta e uma variável de sessão. Com estas informações é criado um objeto da classe `AssistantContext`, que corresponde ao contexto do Assistente, e, em

seguida, é obtida uma instância da classe `NaturalLanguageProcessor`, para a qual são encaminhados o texto da pergunta e o código do idioma.

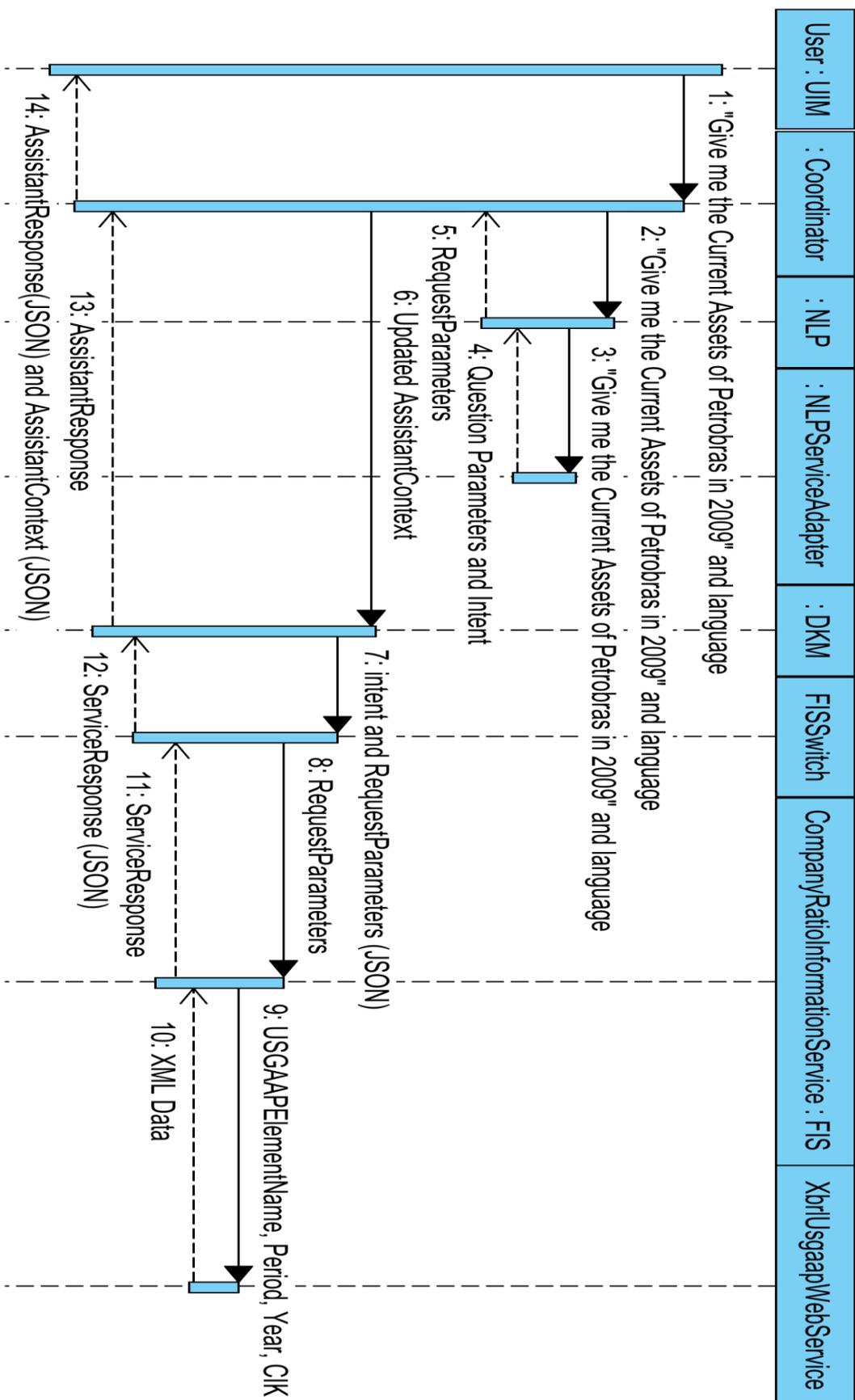
- 3.A instância da classe `NaturalLanguageProcessor` instancia um objeto da classe `NLPServiceAdapter` e encaminha o texto da pergunta e o código do idioma.
- 4.A instância da classe `NLPServiceAdapter` reencaminha estas informações para o web service `NLPProxy`. A classe em Python envia o texto para o web service do NLP da `Api.ai`, inserindo a identificação do agent do NLP da `Api.ai` correspondente ao idioma da consulta. O serviço NLP da `Api.ai` faz o reconhecimento e retorna um objeto de resposta, proprietário, em formato JSON. A classe em Python recebe esse JSON e o encaminha, sem modificações, como resposta para a classe `NLPServiceAdapter`. A classe `NLPServiceAdapter` reconstrói o objeto de resposta do NLP da `Api.ai`, e retorna para a classe `NaturalLanguageProcessor`, a intenção e os parâmetros identificados.
- 5.A classe `NaturalLanguageProcessor` encapsula os parâmetros em um objeto da classe `RequestParameters`, que é retornado para a instância da classe `Coordinator`;
- 6.A instância da classe `Coordinator`, atualiza o objeto da classe `AssistantContext` com o objeto `RequestParameters` recebido e com a intenção. Na sequência ela obtém um objeto da classe `DialogKnowledgeManager`, e encaminha para ele o objeto `AssistantContext` atualizado;
- 7.O objeto da classe `DialogKnowledgeManager` verifica a classe `AssistantContext` e obtém a intenção e os parâmetros da pergunta do usuário. A classe `DialogKnowledgeManager` acessa o web service `FinancialInformationServicesSwitch` enviando os parâmetros nome da intenção e a instância da `RequestParameters` (convertida em JSON).
- 8.A classe `FinancialInformationServicesSwitch`, classe principal do web service, instancia a classe `FinancialInformationServiceFactory` e encaminha a string do nome da intenção para ela, que, em resposta, retorna uma instância da classe `CompanyRatiInformationService`. A classe `FinancialInformationServicesSwitch` reconstrói o objeto

RequestParameters, a partir do JSON recebido, e o submete para a classe CompanyRatioInformationService;

- 9.A instância da classe CompanyRatioInformationService aciona o serviço XbrlUsgaapWebService encaminhando os parâmetros relacionados a tecnologia XBRL e a taxonomia USGAAP, e.g USGAAPElementName, CIK, year e period;
- 10.Com esses parâmetros, o serviço XbrlUsgaapWebService constrói o URL para realizar a consulta ao web service da XBRL-US. Alguns dados de autenticação de uso do web service são incluídos na query de consulta. A classe XbrlUsgaapWebService obtém, como resposta, um documento em XML que é repassado, intacto, como resposta do serviço;
- 11.A instância da classe CompanyRatioInformationService, a partir dos dados em XML recebidos, constrói o texto de resposta, no idioma selecionado, e o embuti em um objeto da classe ServiceResponse, que é enviado como resposta para a classe FinancialInformationServicesSwitch;
- 12.A classe FinancialInformationServicesSwitch converte o objeto ServiceResponse em formato JSON e o envia como resposta;
- 13.A instância da classe DialogKnowledgeManager converte o JSON, recebido do web service, em uma instância da classe ServiceResponse, atualiza o contexto e instancia um objeto da classe AssistantResponse com a resposta do Assistente e retorna para o objeto da classe Coordinator;
- 14.O objeto da classe Coordinator converte os objetos das classes AssistantResponse e AssistantContext em JSON e os encaminha para a aplicação cliente.

Por fim, a aplicação cliente converte o JSON, referente a classe AssistantResponse, e um objeto e obtém o texto "The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00" que é apresentado como resposta no navegador web do usuário.

Figura 34 - Diagrama de Sequência de uma transação bem sucedida



Fonte: Próprio autor.

## 4.6 AMBIENTE DE DESENVOLVIMENTO DO ASSISTENTE

O ambiente de desenvolvimento do Assistente Financeiro Virtual consiste em um conjunto de recursos computacionais utilizados para especificar, codificar, executar, testar e medir os artefatos de software produzidos para essa dissertação.

### 4.6.1 Softwares utilizados

Foram utilizados o Android Studio 1.5.1 fornecido pela Google utilizando o Compile SDK versão Android 5.0 e o Build Tools versão 19.1.0 também fornecidos por ela para implementação da versão cliente móvel compatível com o sistema operacional Android versão igual ou superior a 4.2.

Para a implementação das classes que representam as camadas Orchestration, Understanding / Knowledge e Data, em tecnologia Java, foram utilizadas as seguintes ferramentas:

- NetBeans IDE 8.0.2 Patch 2;
- Java SDK (ORACLE, 2016a) e Java(TM) SE Runtime versão 1.8.0\_72;
- Web Server WebLogic 12.2.1 (ORACLE, 2016c);
- SO: OS X version 10.11.4 - El Capitan - (APPLE, 2016c);
- Navegador Web: Safari 9.1.1 e Firefox 46.0.1 para OS X.

Na implementação do Webservice proxy do NLP foi utilizado o ambiente Python 2.7.10 e as bibliotecas: Flask 0.10.1 (RONACHER, 2016) e Spyne 2.12.9 (ARSKOM, 2016). As bibliotecas foram utilizadas para que o webservice fosse disponibilizado através do protocolo SOAP. Todas ferramentas foram utilizadas com o Sistema Operacional OS X version 10.11.16 (APPLE, 2016c).

Para o Natural Language Processor foi utilizado o *web service* Api.ai da Api.ai e para a camada Data foi utilizado o *web service* da XBRL-US.

### 4.6.2 Hardwares utilizados

Um microcomputador iMac 12,1 com Processador Intel Core i5 2,5 GHz com 4 Núcleos:

- Cache L2 (por Núcleo): 256 KB;
- Cache de L3: 6 MB;
- Memória: 12 GB;
- Armazenamento: 1 SSD de 500GB e 1 HD de 500GB.

Um Tablet:

- Tela de 7”;
- SO Android versão 4.2.2.

#### 4.7 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a implementação das camadas arquiteturais do Assistente Financeiro Virtual, além da justificativa pela opção tecnológica. Quase todos os componentes e serviços foram implementados em Java.

Para a camada Presentation, foram implementados aplicações clientes, uma em javascript / html para ser executada em navegadores web, e outra compatível com o SO móvel Android. Todas elas permitem a interação com o Assistente Financeiro Virtual em dois idiomas, o inglês e o português. A aplicação móvel permite ainda a interação completamente por voz, nos dois idiomas.

Para os componentes da camada Orchestration foram apresentadas as classes principais implementadas. O componente Coordinator foi implementado desacoplado do serviço de processamento de linguagem natural. Além disso, foi criada uma única classe Java para realizar a conversão das informações providas do processador de linguagem natural para o Coordinator, facilitando a substituição do serviço de NLP através da alteração dessa classe.

Já para a camada Understanding / Knowledge, além de algumas classes implementadas, foram apresentadas também a seleção e a configuração do serviço de Natural Language Processing. Para a configuração foi necessária a utilização de uma linguagem proprietária para construção da gramática do serviço de NLP utilizado. Além disso, a fim de reduzir o tempo total para construção da resposta, os significados de alguns termos dos dicionários do NLP foram construídos com um conjunto de informações, encapsuladas em JSON, para que os serviços de dados fossem minimamente acionados.

A configuração do NLP realizada permitiu que o Assistente reconhecesse dois idiomas, o inglês e o português, e determinados termos financeiros conformes com a taxonomia financeira US-GAAP. Os parâmetros reconhecidos pelo NLP, independente de qual dos dois idiomas foi utilizado na pergunta, podem ser utilizados para construir queries de banco de dados ou consulta a *web services* de dados.

O significado composto de cada termo, encapsulado em JSON, possibilita a inserção, para cada elemento da taxonomia financeira utilizada, de um rótulo, ou label, que pode significar uma tradução para o idioma desejado ou um texto de apresentação em um relatório.

Por limitação do serviço de NLP selecionado, foi necessária a codificação de um *web service* em Python, que tem a responsabilidade de encaminhar as perguntas advindas das classes Java para o serviço de NLP e retornar as respostas.

A implementação de um dos serviços de informação financeira foi apresentada, bem como a classe que aciona esses serviços. Essa classe utiliza um *design pattern* do tipo fábrica, que facilita a inclusão de novas classes que representam os serviços de informação financeira. Esses serviços devem previamente ter conhecimento dos parâmetros produzidos pelo NLP, i.e. deve existir um sincronismo entre os serviços da camada Understanding / Knowledge.

Foram mostrados também os serviços de dados em XBRL, pertencentes a camada Data. Foi utilizado um serviço proprietário que disponibiliza dados de demonstrativos financeiros de empresas americanas. Esse serviço aceita como parâmetros nomes de elementos XBRL que seguem a taxonomia US-GAAP. Como resposta, ele retorna um arquivo XML com diversos dados que devem ser manipulados para extrair a informação que se busca.

Por fim, foi apresentado o ambiente de desenvolvimento utilizado no trabalho. Para facilitar a implementação, foi utilizado o IDE NetBeans e o servidor WebLogic, todos gratuitos. A linguagem de programação Java foi primordialmente utilizada na implementação por causa da sua ampla utilização na comunidade de software e pela grande disponibilidade de bibliotecas de software nessa tecnologia para os mais diversos tipos de problema. Todos os componentes e serviços foram implementados no mesmo computador.

No próximo capítulo serão mostrados os testes e a avaliação da implementação do Assistente Virtual Financeiro.

## 5 TESTES E AVALIAÇÃO DO ASSISTENTE FINANCEIRO VIRTUAL

Para testar a eficiência do Assistente Financeiro Virtual foram realizadas perguntas do domínio financeiro para avaliar o Assistente quanto ao entendimento da pergunta, a velocidade da resposta e a sua correção.

### ***5.1.1 Testes de performance do Assistente Financeiro Virtual***

O objetivo desses testes é avaliar a performance do Assistente Financeiro Virtual e identificar o impacto da execução dos serviços no tempo global das respostas.

Para padronizar os resultados e, assim, facilitar as medições e os cálculos, foi criado um script javascript, embutido em uma página web cliente. Esse script submeteu uma sequência de 43 perguntas, escritas em linguagem natural e em inglês, ao Assistente Financeiro Virtual. As perguntas submetidas e a ordem de submissão delas estão apresentadas no apêndice A.

Como premissa, as palavras chaves inseridas nas perguntas dessa sequência foram previamente registradas nos dicionários de termo e todas as perguntas seguiam os padrões registrados na gramática do Natural Language Processor, i.e. o nome das empresas, os termos financeiros, as estruturas das perguntas, já faziam parte do domínio de conhecimento do Assistente, antes da execução do teste. O objetivo principal desse prévio conhecimento foi evitar um eventual não acionamento do serviço de provimento de dados o que provocaria uma diminuição nos tempos de execução da consulta, interferindo nos resultados finais das aferições de tempo de resposta do Assistente.

Os tempos de execução e resposta foram aferidos com o mesmo grupo de perguntas em dias diferentes e em turnos distintos do dia, a fim de dirimir os efeitos momentâneos provocados por congestionamentos da rede, sobrecarga dos serviços utilizados, ou processamentos nos computadores clientes e/ou servidores.

Os testes foram realizados entre os dias 7 de junho de 2016 e 16 de agosto, do mesmo ano, em horários distintos e resultou em 16 amostras. Uma planilha de dados com os resultados das medições realizadas está ilustrada no apêndice C.

Para cada amostra foram medidos os tempos de desempenho individual dos serviços utilizados e o desempenho total do Assistente Financeiro Virtual, conforme apresentado a seguir.

1. Tempo de resposta total do Assistente;
2. Tempo de resposta do NLP da Api.ai;
3. Tempo de resposta do serviço de dados da XBRL-US;

Para obter os valores da primeira medição, o mesmo script na página web cliente que submete a sequência de perguntas ao Assistente também registra o tempo gasto em obter a resposta de cada pergunta. No apêndice B está apresentada uma amostra de uma medição realizada pelo navegador web.

Para obter os valores da segunda medição, o script em Python que funciona como um proxy do *web service* do NLP da Api.ai, também registra o tempo gasto individual entre a submissão de uma pergunta ao serviço de NLP Api.ai e a chegada da resposta correspondente.

Para obter os valores da terceira medição, a mesma classe Java que aciona o *web service* de dados da XBRL-US, também registra o período de tempo despendido entre a submissão da query e a chegada do resultado.

### **5.1.2 Resultados dos testes de performance**

A média global do tempo de resposta do assistente para às perguntas que ele entendeu foi em torno de 2 segundos. Aproximadamente 42% do tempo para o Assistente Financeiro Virtual construir a resposta foram de responsabilidade do processamento do *web service* Natural Language Processor da Api.ai e da comunicação com o serviço. Já o tempo gasto com o processamento do serviço de provimento de dados da XBRL-US, e com a comunicação com o serviço correspondeu a 55,45% do tempo de construção da resposta, conforme dados apresentados na Tabela 1.

Com o objetivo de avaliar também uma possível utilização do recurso de *cache* pelos serviços avaliados, e o efeito sobre tempo de resposta deles, duas das perguntas da sequência submetida ao Assistente foram realizadas repetidamente por cinco vezes, cada uma. Percebeu-se que as perguntas enviadas repetidamente influenciaram significativamente no tempo médio global das respostas do assistente, que fez diminuir praticamente 700 milissegundos, e resultou num tempo médio de resposta global aproximadamente igual a 1,3 segundos, conforme apresentado na Tabela 1. Este resultado evidenciou o uso do recurso de *cache* e o efeito positivo

que ele proporcionou. Constatado isso, foi necessário identificar quais os serviços utilizados usaram o recurso e qual a influência deste uso em cada serviço.

Tabela 1 - Tempos de respostas do Assistente Financeiro Virtual

Escopo da medição	Média global do Assistente	Média do NLP do Api.ai	Média do Serviço Xbrl US	% do NLP do Api.ai	% do Serviço Xbrl US	Soma dos serviços	% Soma dos serviços
Todas as 43 perguntas	2011,15	848,29	1115,21	42,18	55,45	1963,50	97,63
Somente perguntas repetidas	1291,35	833,30	412,83	64,53	31,97	1246,13	96,50
Diferença	719,79	14,99	702,38	-22,35	23,48	717,37	1,13

Fonte: Próprio autor.

Nota: Unidade de medida – milissegundos.

Foi percebido que apesar dos tempos terem diminuído com a submissão de palavras repetidas, o tempo de resposta do serviço de Natural Language Processor da Api.ai foi relativamente o mesmo, entretanto, a responsabilidade sobre o tempo total aumentou para 64,53%. Já o serviço de provimento de dados da XBRL-US teve uma redução média de quase 700 milissegundos, semelhante a redução no tempo total do Assistente, e a sua responsabilidade sobre esse tempo foi reduzida para 31,97%, que corresponde a uma redução de 23,48%, conforme apresentado na Tabela 1. Com esses dados, ficou evidenciado o benefício positivo para os tempos de resposta do Assistente, alcançados através do uso do cache em um desses serviços, além de indicar que o único serviço que o utilizou foi o de provimento de dados.

Como os dois serviços citados foram responsáveis por 97,63% do tempo total gasto para que o Assistente respondesse as perguntas, uma melhora no desempenho do processamento e na comunicação de ambos os serviços, diminuirá, significativamente, o tempo de resposta do Assistente Financeiro Virtual. O uso de cache num dos serviços comprovou a eficácia do recurso para a redução sensível do tempo para responder as perguntas repetidas. Não foi possível medir o tempo gasto com a comunicação utilizada para acessar os serviços e assim diagnosticar qual o tempo efetivamente consumido pelos respectivos processamentos.

Ocorreram alguns episódios de demora do Assistente Financeiro Virtual em responder as questões, provocados por lentidão no serviço de dados da XBRL-US em determinados horários do dia. Eles foram os principais responsáveis pelo pior

tempo de resposta do Assistente nos testes, que foi de 11,17 segundos. O melhor tempo de resposta obtido pelo Assistente foi de 0,93 segundos.

### **5.1.3 Teste comparativo com os assistentes virtuais de mercado**

Os assistentes virtuais de mercado utilizados para o teste comparativo foram o assistente virtual de propósito geral da Apple, o Siri, e o Google Now da Google. Eles foram selecionados por estar instalados em mais de 90% dos smartphones disponíveis no mundo.

Para realizar o teste comparativo, foi utilizada a mesma sequência de perguntas em inglês submetidas ao Assistente Financeiro Virtual, entretanto a sequência foi submetida uma única vez a cada um deles.

Cada pergunta foi inicialmente realizada por voz para reconhecimento em inglês, e nos casos de incorreção provocados por falhas no reconhecimento de algum termo da pergunta, a pergunta era editada através da digitação, e submetida novamente.

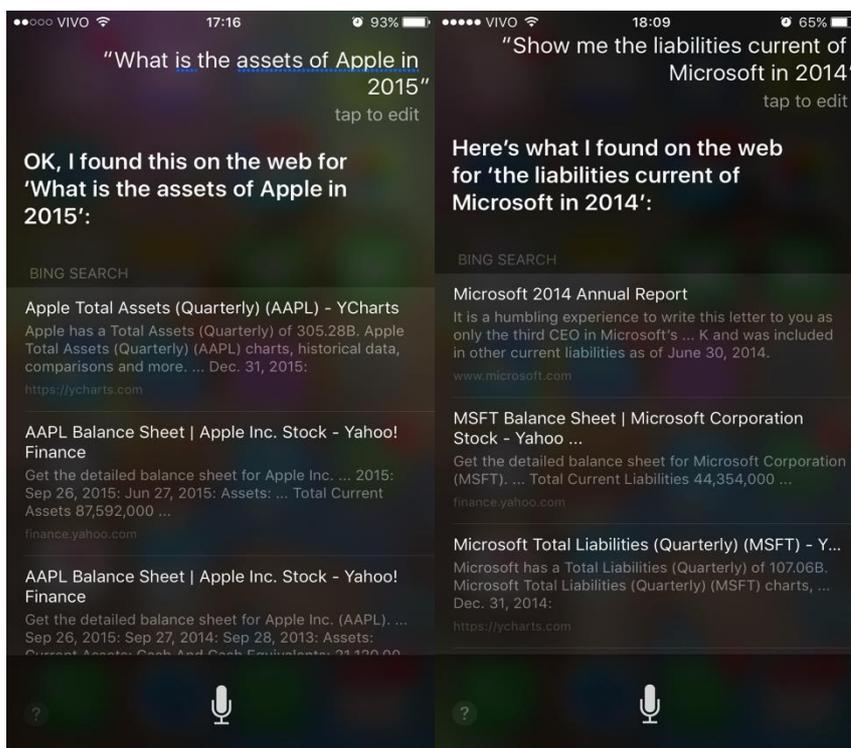
O objetivo desse teste foi analisar comparativamente a correção das respostas financeiras e a realização da manutenção do diálogo feitas pelos dois assistentes.

### **5.1.4 Resultado do teste comparativo**

Para 27 das 43 perguntas, o Siri mostrou na tela do dispositivo, em que estava sendo executado, a resposta fornecida pelo serviço de busca Bing. A resposta consistia em uma lista de links de páginas na web, que continham informações relacionadas as perguntas financeiras, conforme ilustrado na Figura 35. Além de mostrar os links, o Siri emitiu frases informando que havia encontrado as informações na web, por exemplo com as seguintes frases, *“Here is what I found on the web for ...”*, ou *“OK, I found this on the web for ...”*.

Para todas as 43 perguntas, o Google Now mostrou na tela do dispositivo, em que estava sendo executado, a resposta fornecida pelo serviço de busca do Google. As respostas consistiam em uma lista de links de páginas na web, que continham informações relacionadas as perguntas financeiras. Não foi emitida nenhuma voz. Neste caso ele atuou com um simples assistente de digitação ao serviço de busca do Google.

Figura 35 - Resultados do Assistente Virtual Siri da Apple

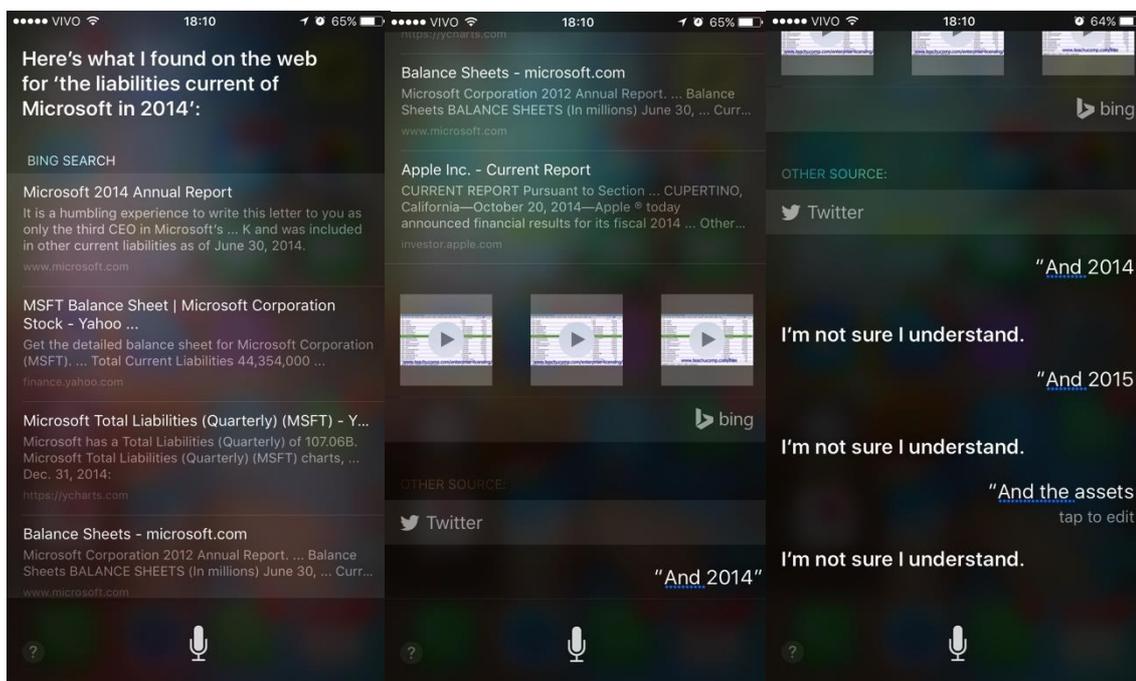


Fonte: Próprio autor.

Das 43 perguntas submetidas, 10 foram perguntas curtas e complementares, isto é, elas aproveitavam a pergunta anterior e alterava apenas um parâmetro da questão. O objetivo deste teste foi verificar se o Siri simulava um diálogo entre ele e o usuário, ao responder perguntas curtas que levam em conta as informações passadas ao assistente em perguntas anteriores. Todas as 10 perguntas foram tratadas pelo Siri como se fossem novas perguntas, o assistente virtual não considerou as informações passadas pelas perguntas anteriores.

Analisando, como exemplo, uma amostra de uma sequência de 4 perguntas submetidas ao Siri: após o Siri responder a pergunta "*Show me the liabilities current of Microsoft in 2014*" com uma lista de links, submeteu-se ao Siri outra pergunta "*and in 2014*", solicitando o resultado em 2014. O Siri respondeu que não havia entendido o que foi perguntado. Em sequência outras perguntas curtas foram realizadas, e.g. "*and in 2014*" e "*and the assets*", que também não foram entendidas pelo Siri, conforme Figura 36.

Figura 36 – Sequência de respostas do Siri para perguntas curtas incompletas



Fonte: Próprio autor.

Todas as 10 perguntas curtas submetidas ao Google Now, para verificar se ele conseguia simular um diálogo com o usuário, foram tratadas pelo Google Now como se fossem perguntas novas, assim como o Siri, e as respondeu com uma lista de links sem considerar o contexto do diálogo.

Essa mesma sequência de perguntas curtas foi submetida ao Assistente Financeiro Virtual e foi entendida e respondida corretamente por ele, conforme a seguir:

1. Pergunta submetida: “Show me the liabilities current of Microsoft in 2014”;
2. Resposta do Assistente Financeiro: “The Current Liabilities of MICROSOFT CORPORATION Company in 2014 is US\$45,625.00”;
3. Pergunta submetida: “and 2014”;
4. Resposta do Assistente Financeiro: “The Current Liabilities of MICROSOFT CORPORATION Company in 2014 is US\$45,625.00”;
5. Pergunta submetida: “and 2015”;
6. Resposta do Assistente Financeiro: “The Current Liabilities of MICROSOFT CORPORATION Company in 2015 is US\$49,647.00”;
7. Pergunta submetida: “and the assets”;

8.Resposta do Assistente Financeiro: “The Assets of MICROSOFT CORPORATION Company in 2015 is US\$174,472.00”.

Este comportamento indica que nem o Siri e nem o Google Now mantêm um diálogo com o usuário, semelhante a uma conversa entre pessoas, na área financeira, área do conhecimento das quais as perguntas foram realizadas.

Nesse caso, o Assistente Financeiro Virtual entendeu todas as perguntas enviadas em sequência, manteve o diálogo e respondeu com correção todas as perguntas.

## 5.2 CONSIDERAÇÕES FINAIS

Os resultados do Assistente Financeiro Virtual registrados apresentaram um tempo médio de resposta igual a 2 segundos. Esses resultados são compatíveis com os assistentes mais utilizados pelo mercado, e.g. Siri, Google Now. Todas as perguntas financeiras submetidas foram entendidas e respondidas corretamente pelo Assistente Financeiro Virtual, diferente do que aconteceu com os assistentes virtuais de mercado, que para a maioria das respostas forneceu apenas uma lista de links para sites web financeiros. Algo justificável, pois tanto o Siri, quanto o Google Now, ainda não contemplam o domínio financeiro.

A avaliação da gestão do diálogo com usuário, implementada no Assistente Financeiro Virtual também foi positiva, pois o Assistente entendeu a intenção e respondeu corretamente as perguntas curtas que levavam em conta informações passadas em perguntas anteriores. Nesta questão o Assistente Financeiro Virtual foi melhor do que os dois assistentes de mercado, que não entenderam corretamente nenhuma das perguntas curtas realizadas.

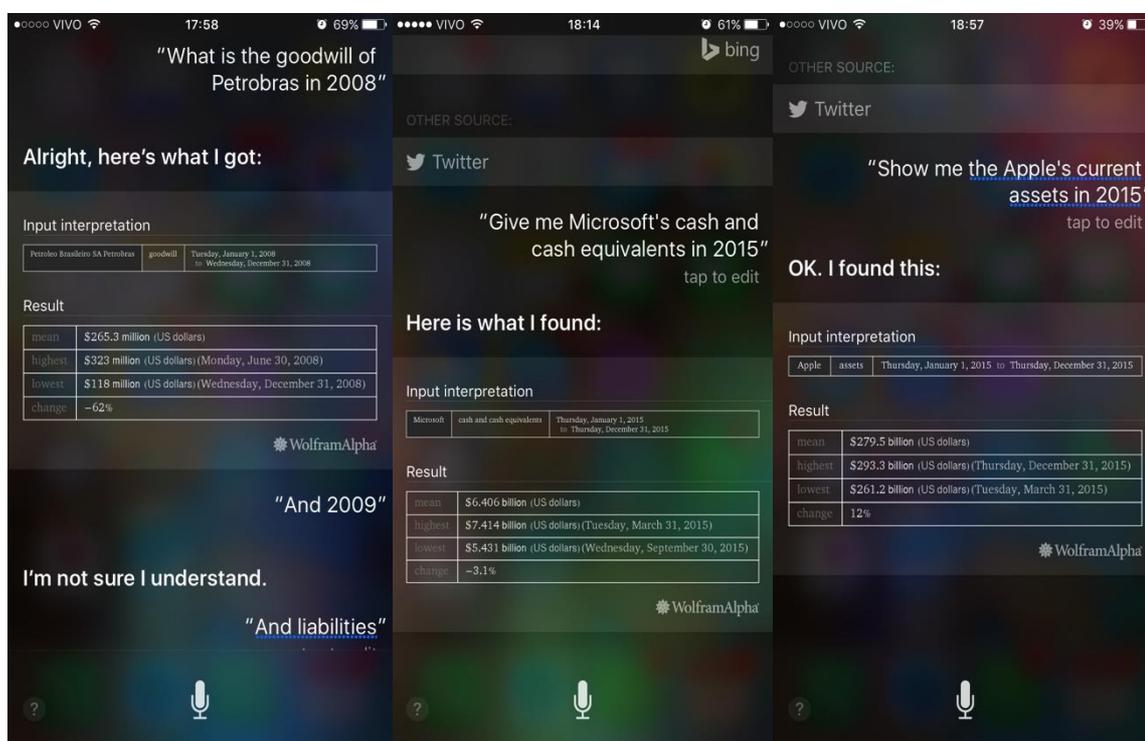
Os resultados obtidos com os tempos de resposta, a manutenção do diálogo e a correção das informações disponibilizadas corroboram com a viabilidade do Assistente Financeiro Virtual.

Apesar do resultado desfavorável para o Siri em relação a trazer a informação solicitada, um fato relevante ocorreu durante a execução dos testes. Em 6 das 43 perguntas submetidas, o Siri, mesmo sendo um assistente de propósito específico, conseguiu identificar todos os parâmetros financeiros inseridos nas perguntas, como: a empresa, o item financeiro e o período. Esta identificação foi obtida com o suporte do serviço chamado WolframAlpha (WOLFRAM, 2016b) que disponibiliza

informações em determinadas áreas do conhecimento mediante consultas feitas em linguagem natural. Essa integração acontece porque foi realizado um convênio entre a Apple e a empresa Wolfram (2016a), a fornecedora do serviço.

Exemplos de algumas perguntas identificadas: “*What is the goodwill of Petrobras in 2008*”, “*Give me the Microsoft’s cash and cash equivalents in 2015*” e “*Show me the Apple’s current assets in 2015*”, conforme ilustrado na Figura 37. Não foi possível identificar a razão do serviço não ter reconhecido os parâmetros das outras perguntas realizadas.

Figura 37 – Perguntas que o Siri identificou os parâmetros financeiros



Fonte: Próprio autor.

Este fato indica que, apesar de ser um assistente de propósito geral, o Siri, ao submeter a pergunta a um serviço especializado no domínio do conhecimento, consegue identificar corretamente as palavras chaves do domínio financeiro e responder com mais proximidade ao que o usuário perguntou. Ainda cabe melhorias nesse recurso, pois apesar de ter reconhecido os parâmetros financeiros dessas seis perguntas, o Siri forneceu uma faixa de valores próximos ao valor esperado como resposta. O valor exato perguntado não foi informado, conforme exemplos ilustrados na Figura 37.

## 6 CONCLUSÃO

Nesta dissertação foi apresentado um sistema computacional de assistência ao usuário para realizar consultas de informações financeiras, baseado em dados descritos a partir da tecnologia XBRL, que utiliza linguagem natural como forma de interação com o usuário. Este sistema, nomeado Assistente Financeiro Virtual, é um assistente virtual do tipo pergunta/resposta que reconhece diversos questionamentos, feitos no idioma inglês ou no idioma português, sobre a situação de indicadores financeiros de algumas empresas que disponibilizam suas respectivas demonstrações financeiras em formato eletrônico. Para trazer as respostas o usuário deve informar ao sistema, através das perguntas, o nome da empresa, o ano ou o período contábil e o conceito financeiro, utilizado em análises financeiras, baseado na taxonomia US-GAAP.

As perguntas do usuário e as respostas do Assistente podem ser realizadas por voz. O diálogo com o usuário é mantido pelo sistema através da administração do contexto da sessão de perguntas, e consegue responder com sucesso algumas perguntas curtas que levam em conta as informações passadas em questionamentos prévios do usuário.

Todos os requisitos definidos para o Assistente Financeiro Virtual na Seção 1.2 foram suportados. As considerações a respeito de cada requisito estão apresentadas no Quadro 12.

Quadro 12 - Considerações sobre os requisitos do Assistente Financeiro Virtual

Requisito	Consideração
a	A sequência de respostas fornecidas pelo Assistente às perguntas realizadas nos testes do Capítulo 5 evidenciou a conformidade com o requisito.
b	As perguntas feitas nos testes do Capítulo 5 utilizaram linguagem natural. Somente parte do jargão financeiro que foi implementado no Assistente foi entendido por ele.
c	Conforme evidenciado e comentado no Capítulo 5.
d	O Assistente utiliza os dados públicos financeiros fornecidos pela US-SEC, conforme apresentado na Seção 4.4.1 .
e	Conforme a arquitetura apresentada no Capítulo 3.

f	Conforme tecnologias implementadas nas camadas do Assistente apresentadas no Capítulo 4.
---	--

Este trabalho buscou responder as questões de pesquisa apresentadas na Seção 1.3. As questões e as suas respectivas respostas serão apresentadas a seguir.

**1.Os assistentes virtuais disponíveis atualmente podem ser customizados para responder questões de um domínio específico?**

R1. Foram encontrados alguns assistentes virtuais e serviços de processamento de linguagem natural que podem ser customizados para um domínio específico, e.g. Sys Virtual Assistant, Api.ai. Na comunidade acadêmica foram encontrados alguns trabalhos que apresentaram implementações desse tipo de assistente, conforme apresentados no Capítulo 2.

**2.Os *frameworks* ou arquiteturas disponíveis permitem gerar assistentes virtuais de domínio restrito do tipo pergunta e resposta?**

R2. Foram encontrados alguns trabalhos, como o *framework* de Eisman et al (2012), e as arquiteturas de Paraíso e Barthés (2005), e de Zambiasi e Rabelo (2012) que podem gerar assistente virtuais do tipo pergunta e resposta. Entretanto, cada um deles tem características e restrições distintas que devem ser conhecidas para avaliar se as especificações do assistente, que se quer criar, são suportadas por eles, conforme apresentado no Capítulo 2.

**3.O Assistente Financeiro Virtual pode ser totalmente baseado em uma arquitetura orientada a serviço, inclusive utilizando um serviço de processamento de linguagem natural?**

R3. A implementação e os testes apresentados neste trabalho, comprovam que o Assistente Financeiro Virtual pode ser totalmente baseado em serviços, até mesmo com o desacoplamento total da unidade de processamento de linguagem natural, algo que não foi projetado em nenhuma das arquiteturas e *frameworks* apresentados em trabalhos na comunidade acadêmica que se teve acesso.

**4.É possível realizar consultas de informações financeiras em dados disponibilizados em XBRL utilizando linguagem natural? Em caso positivo, essa mesma consulta de dados disponibilizados em XBRL pode ser realizada em idiomas diferentes?**

R4. Através do recurso de processamento de linguagem natural, é possível converter perguntas feitas nos mais diversos idiomas em linguagem natural, e extrair delas parâmetros padronizados, compatíveis com a tecnologia XBRL, e independentes do idioma. Esses parâmetros podem ser utilizados para construir queries de banco de dados XBRL ou para acessar serviços que disponibilizam dados disponibilizados em formulários eletrônicos em XBRL.

**5.O Assistente Financeiro Virtual pode realizar análises financeiras complexas utilizando as informações registradas em relatórios em XBRL?**

R5. A versão do Assistente Financeiro Virtual, implementada para este trabalho, não faz este tipo de análise pois nenhuma classe foi codificada para tal, entretanto, é plenamente possível para o Assistente realizar estas análises com uma manutenção de software sem interferências arquiteturais. A adição de novas respostas e comportamentos está prevista na arquitetura, proposta neste trabalho, que foi utilizada como base para a implementação. Para que Assistente Financeiro Virtual realize este tipo de análise, é necessária uma intervenção na camada Understanding / Knowledge, que consiste em implementar um serviço de informação financeira para fornecer estas análises financeiras complexas e configurar a gramática e o dicionário do processador de linguagem natural para capturar os parâmetros compatíveis com esse novo serviço. Esse novo serviço poderá obter os dados do serviço já implementado para este trabalho, na camada Data.

**6.O Assistente Financeiro Virtual pode utilizar taxonomias financeiras diferentes?**

R6. A versão do Assistente Financeiro Virtual, implementada para este trabalho, é compatível com a taxonomia US-GAAP. Entretanto, a utilização de outras taxonomias é prevista na arquitetura dele, e para isso, é necessário intervir nas camadas Understanding / Knowledge e Data, através da configuração do dicionário do processador de linguagem natural e da

implementação, ou uso, de um serviço, ou banco, de dados que disponibilize os dados disponibilizados na taxonomia desejada.

**7.Os serviços de processamento de linguagem natural disponíveis atualmente possuem tempos de resposta que permitem a construção de assistentes virtuais com tempos de resposta compatíveis com os assistentes virtuais disponíveis no mercado?**

R7. Para este trabalho, apenas um dos serviços de processamento de linguagem natural foi utilizado e medido. Com o uso dele, o Assistente Financeiro Virtual obteve tempos médios de respostas compatíveis com os assistentes virtuais disponíveis no mercado, e.g. Siri, Google Now, que são inferiores a 2 segundos. Entretanto, registrou-se tempos médios de resposta semelhantes a 850 milissegundos para o serviço utilizado, resultado que pode inviabilizar projetos de assistente virtuais que acessem muitos serviços e que tenham que responder as perguntas em menos de 3 segundos.

Os resultados do Assistente Financeiro Virtual apresentados demonstraram um tempo médio de resposta igual a 2 segundos. Esses resultados são compatíveis com os assistentes virtuais disponíveis no mercado, que foram utilizados neste trabalho, e.g. Siri, Google Now, apesar da ocorrência de alguns episódios de demora do Assistente Financeiro Virtual em responder algumas questões durante os testes. Esses atrasos foram provocados por lentidão no serviço de extração de dados em XBRL utilizado pelo Assistente, conforme avaliação apresentada na Seção 5.1.2 .

Para evitar lentidões nas respostas, evitou-se acessos desnecessários ao serviço da XBRL-US. Um dos recursos foi embutir, em formato JSON, um conjunto de informações mínimas necessárias às consultas financeiras nos dicionários do Natural Language Processor. Assim aproveita-se o acesso ao NLP para obter essas informações e evita-se a busca das mesmas através do serviço da XBRL-US, conforme apresentado na Seção 4.3.2 .

O desacoplamento do serviço do Natural Language Processor, na implementação do Coordinator, da camada Orchestration, além de seguir a arquitetura proposta, facilitou a substituição do NLP, pois somente é necessária a alteração em uma única classe, que é a classe que faz a conversão dos dados

advindos do serviço de NLP para os parâmetros esperados pela Coordinator, conforme apresentado na Seção 4.2.4 .

Todas as respostas submetidas ao Assistente Financeiro Virtual foram entendidas e respondidas corretamente por ele, diferente do que aconteceu com o Siri e o Google Now, que para a maioria das respostas forneceram apenas uma lista de links para web sites financeiros. Algo justificável, pois nem o Siri e nem o Google Now contemplam o domínio financeiro.

A avaliação da gestão do diálogo com o usuário feita pelo Assistente Financeiro Virtual também foi positiva, pois ele respondeu corretamente as perguntas curtas que levavam em conta informações passadas em perguntas anteriores. Nesta questão o Assistente Financeiro Virtual foi melhor do que os dois assistentes avaliados, que não entenderam corretamente nenhuma das perguntas curtas realizadas em sequência.

A avaliação positiva dos resultados obtidos com os testes de desempenho, correção das respostas disponibilizadas e a manutenção do diálogo com o usuário, confirmam a viabilidade do Assistente Financeiro Virtual, como um assistente de pergunta e resposta que auxilia o usuário, com suporte opcional a voz, através de linguagem natural, a obter informações sobre os índices, ou conceitos, financeiros de determinadas empresas.

## 6.1 PRINCIPAIS CONTRIBUIÇÕES

Após a revisão da literatura; a escrita de um artigo, que foi aceito em uma conferência internacional e publicado em uma biblioteca digital; a implementação completa da ideia proposta e a realização dos testes, entende-se que o desenvolvimento deste trabalho trouxe as seguintes contribuições:

- Um estudo sobre um sistema de assistência ao usuário de consultas de informações financeiras utilizando linguagem natural e processamento de voz através de serviços que acessam uma base de dados em XBRL;
- A proposição de uma arquitetura para assistentes de propósito específico, que pode ser utilizada para construir assistentes virtuais do tipo pergunta/resposta para domínios específicos do conhecimento com suporte a reconhecimento de perguntas realizadas em linguagem natural em diversos idiomas;

- A análise comparativa de alguns serviços de processamento de linguagem natural disponíveis no mercado, que serviu para selecionar o serviço de processamento de linguagem natural que foi utilizado pelo Assistente Financeiro Virtual;
- A implementação de um assistente virtual multilíngue, que responde à determinadas questões financeiras relacionadas a empresas, suportada por uma base de dados XBRL;
- Uma avaliação da gestão de diálogo com o usuário, implementada no Assistente Financeiro Virtual, que permite ao Assistente reconhecer, e até mesmo responder, com sucesso perguntas curtas do usuário que foram feitas em sequência durante o diálogo entre eles. Entende-se por perguntas curtas, as perguntas que não possui todos os parâmetros necessários para uma consulta financeira. Essa funcionalidade não foi identificada nos assistentes virtuais disponíveis no mercado, que foram avaliados nos testes neste trabalho, e.g. Siri, Google Now;
- Uma avaliação do desempenho do Assistente Financeiro Virtual: o resultado dos testes com uma implementação, que utilizou serviços fornecidos por terceiros, sinalizou a viabilidade da arquitetura e da integração das tecnologias utilizadas, como processamento de linguagem natural e XBRL. Indicou também a responsabilidade direta dos serviços de dados e de processamento de linguagem natural, em torno de 97%, sobre os tempos totais obtidos. Além evidenciar o benefício do uso do recurso de cache nos serviços de extração de dados XBRL que contribuiu para uma sensível redução do tempo de resposta.

## 6.2 PRINCIPAIS RESTRIÇÕES E LIMITAÇÕES

Durante o desenvolvimento deste trabalho algumas dificuldades foram encontradas, e apesar das contribuições obtidas há algumas restrições e limitações deste trabalho, conforme a seguir:

Os componentes e serviços de processamento de linguagem natural disponíveis no mercado não são muitos e a maioria deles são proprietários. A escassez desses serviços e componentes trouxe algumas dificuldades no processo de seleção e utilização deles. Houve a necessidade de conhecer a metodologia e a linguagem proprietária para configuração do dicionário e da gramática para

identificar a facilidade de utilização e a aderência a proposta do Assistente Financeiro Virtual.

O Assistente Financeiro Virtual não reconhece qualquer pergunta financeira que não esteja configurada para ser entendida. As perguntas realizadas nos testes foram compulsoriamente escolhidas de acordo com as configurações feitas nas gramáticas e nos dicionários do processador de linguagem natural. Perguntas que não são esperadas pelo Assistente não trazem nenhuma informação relevante para o usuário, somente um aviso que ele não entendeu a pergunta.

A versão implementada não permite a adição de novas empresas ou novos termos financeiros pelo próprio usuário, apesar da arquitetura permitir este tipo de operação. A classe relacionada ao contexto do Assistente foi implementada para suportar esta funcionalidade, entretanto, as classes que fazem a atualização dos dicionários do processador de linguagem natural não foram implementadas.

A versão do Assistente implementada está configurada para obter valores de relatórios que estejam estruturados em documentos eletrônicos compatíveis com a taxonomia US-GAAP e a tecnologia XBRL. Não foram encontrados serviços ou repositórios de dados compatíveis com outra taxonomia que estivessem disponíveis para que o Assistente pudessem consultar.

Apesar da implementação do Assistente Financeiro Virtual, apresentada neste trabalho, acessar os dados XBRL através de um serviço de dados, é plenamente possível acessar qualquer base de dados XBRL, via web ou local. Para isso, é necessária implementar uma classe que construa as queries que serão submetidas ao banco de dados. O acesso a uma base de dados incompatíveis com XBRL também é possível, entretanto é necessário, além implementar a classe citada anteriormente deve-se criar um novo dicionário de termos compatível com a terminologia e/ou taxonomia desejadas.

### 6.3 PUBLICAÇÃO

Durante a realização deste trabalho, um artigo científico foi submetido a um evento internacional, com o objetivo de obter a validação da comunidade acadêmica de algumas das contribuições. Assim, a comunidade científica pode avaliar e contribuir para o desenvolvimento desta dissertação. Os dados da identificação do trabalho completo que foi publicado corresponde a:

- ABRAAO, Adalberto Alves; SILVA, Paulo Caetano da. **A XBRL Financial Virtual Assistant**. In: ICIW 2016 - The Eleventh International Conference on Internet and Web Applications and Services, 2016, Valencia, Spain, ISSN: 2308-3972, ISBN: 978-1-61208-474-9, p. 64-72.

#### 6.4 TRABALHOS FUTUROS

Após o desenvolvimento deste trabalho, as seguintes idéias e sugestões para trabalhos futuros a fim de evoluir o Assistente Financeiro Virtual serão apresentadas:

Implementar uma funcionalidade que permita ao Assistente Financeiro Virtual ampliar o seu conhecimento financeiro através da interação com o usuário, permitindo a introdução de novos conceitos financeiros e novas empresas.

Implementar um componente na camada Data, integrado com o processador de linguagem natural que permita fazer a seleção da taxonomia e da fonte de dados, o que permitirá que o Assistente acesse simultaneamente dados em taxonomias distintas.

A implementação de um componente que permita que a gramática de qualquer processador de linguagem natural que seja utilizado pelo Assistente Financeiro Virtual seja configurada através de uma linguagem padrão para construção de gramáticas, e.g. JSpeech Grammar Format, eliminando a necessidade de aprendizagem de nova linguagem ou metodologia proprietária para este fim.

A fim de facilitar ainda mais a ampliação do conhecimento do Assistente, justifica-se a criação de um processador de linguagem natural que reconheça as perguntas do usuário e extraia os parâmetros financeiros sem a necessidade prévia de configurar os padrões de pergunta.

## REFERÊNCIAS

ABRAÃO, Adalberto Alves; SILVA, Paulo Caetano da. A XBRL Financial Virtual Assistant. In: ICIW 2016 - THE ELEVENTH INTERNATIONAL CONFERENCE ON INTERNET AND WEB APPLICATIONS AND SERVICES, 11., 2016, Valencia, Spain. **ICIW2016**. [s. l.]: Iaria Xps Press, 2016. p. 64 - 72. Disponível em: <[http://www.thinkmind.org/download.php?articleid=iciw\\_2016\\_4\\_20\\_20039](http://www.thinkmind.org/download.php?articleid=iciw_2016_4_20_20039)>. Acesso em: 11 set. 2016.

API.AI. **Api.ai**. 2016. Disponível em: <<https://api.ai>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Assistant.ai**. 2016. Disponível em: <<https://assistant.ai>>. Acesso em: 11 set. 2016.

APPLE. **Apple**. 2016. Disponível em: <<http://www.apple.com>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **IOS**. 2016. Disponível em: <<http://www.apple.com/ios/>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **OS X**. 2016. Disponível em: <<http://www.apple.com/osx>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **SIRI**. 2016. Disponível em: <<http://www.apple.com/siri>>. Acesso em: 11 set. 2016.

ARSKOM. **Spyne**: RPC that doesn't break your back. Disponível em: <<http://spyne.io>>. Acesso em: 11 set. 2016.

CHIHANI, Bachir; BERTIN, Emmanuel; CRESPI, Noël. A user-centric context-aware mobile assistant. In: INTERNATIONAL CONFERENCE ON INTELLIGENCE IN NEXT GENERATION NETWORKS (ICIN), 17., 2013. **Proceedings...** 2013. p. 110-117. Disponível em: <<http://ieeexplore.ieee.org/document/6670901/>>. Acesso em: 11 set. 2016.

ECMA INTERNATIONAL. **ECMA-404**: The JSON Data Interchange Format. 1 ed. Geneva: Ecma International, 2013. 8 p. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Introducing JSON**. Disponível em: <<http://www.json.org>>. Acesso em: 11 set. 2016.

EISMAN, Eduardo M.; LÓPEZ, Víctor; CASTRO, Juan Luis. A framework for designing closed domain virtual assistants. **Expert Systems With Applications**. Granada, Spain, v. 39, n. 3, p. 3135-3144. 15 mar. 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S095741741101308X>>. Acesso em: 11 set. 2016.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. **Métodos de pesquisa**. Porto Alegre: Editora da Ufrgs, 2009. 120 p. (Série Educação a Distância). ISBN 978-85-386-0071-8. Disponível em: <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso em: 11 set. 2016.

GERON, Cecília Moraes Santostaso; BITTENCOURT, Roberta E.; RICCIO, Edson Luiz. O uso da linguagem XBRL pelas companhias brasileiras. **Práticas em Contabilidade e Gestão**. São Paulo, v. 1, n. 1, p. 117-146, dez. 2013. Disponível em: <<http://editorarevistas.mackenzie.br/index.php/pcg/article/view/6534>>. Acesso em: 11 set. 2016.

GOOGLE. **Google**. 2016. Disponível em: <<http://www.google.com>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Android OS**. 2016. Disponível em: <[https://www.android.com/intl/en\\_us/](https://www.android.com/intl/en_us/)>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Android Studio**. 2016. Disponível em: <<https://developer.android.com/studio/index.html/>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Google Chrome**. 2016. Disponível em: <<https://www.google.com/chrome/>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Google Now**. 2016. Disponível em: <<https://www.google.com/now/>>. Acesso em: 11 set. 2016.

GRUBER, Tom. **Biography**. 2015. Disponível em: <<http://tomgruber.org/bio/short-bio.htm>>. Acesso em: 11 set. 2016.

IFRS. **IFRS home**. 2016. Disponível em: <<http://www.ifrs.org>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **International Accounting Standards Board (IASB)**. 2016. Disponível em: <<http://www.ifrs.org/About-us/IASB/Pages/Home.aspx>>. Acesso em: 11 set. 2016.

YAHOO. **Yahoo**. 2016. Disponível em: <<https://www.yahoo.com>>. Acesso em: 11 set. 2016.

JACKSON PROJECT HOME. **Jackson - Java JSON API**. 2016. Disponível em: <<https://github.com/FasterXML/jackson>>. Acesso em: 11 set. 2016.

KAVITHA, G; PANDE, Sanjay. Review of Dialogue Systems' Methodologies. In: INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND APPLICATIONS (ICISA), 2013. **Proceedings...** 2013. p.1-4, 2013. IEEE. 18.08.2013. DOI 10.1109/ICISA.2013.6579437 ISBN 978-1-4799-0604-8. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6579437>>. Acesso em: 11 set. 2016.

LANGE, Danny B. Mobile objects and mobile agents: The future of distributed computing?. **ECOOOP'98 - Object-Oriented Programming**. Springer-Verlag Berlin Heidelberg. 1998. p. 1-12. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.7189&rep=rep1&type=pdf>>. Acesso em: 11 set. 2016.

LASATER, Christopher G.. **Design Patterns**. Texas: Wordware Publishing, Inc., 2007. 285 p. ISBN-13: 978-1-59822-031-5.

MARTIN, Paul et al. SpeechActs: a spoken-language framework. **Computer**, v. 29, n. 7, p. 33-40, jul. 1996. IEEE. 06.08.2002. DOI 10.1109/2.511966. ISSN: 0018-9162. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=511966>>. Acesso em: 10 jul. 2016

MEDINA, Javier; EISMAN, Eduardo M.; VIRTUAL, Juan Luis Castro. Asistentes virtuales en plataformas 3.0. **Revista Iberoamericana de Informática Educativa**, Granada, Spain, v. 18, p.41-49, dez. 2013. n. 18, ISSN: 1699-4574. Disponível em: <<http://dialnet.unirioja.es/descarga/articulo/4468692.pdf>>. Acesso em: 10 jul. 2016.

MELHORAMENTOS. **Michaelis Dicionário Brasileiro da Língua Portuguesa**. [S. l.]: Melhoramentos Ltda, 2016. Disponível em: <<http://michaelis.uol.com.br/>>. Acesso em: 11 set. 2016.

MICROSOFT. **Bing**. 2016. Disponível em: <<http://www.bing.com/>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Cortana**. 2016. Disponível em: <<https://developer.microsoft.com/en-us/cortana>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Microsoft**. 2016. Disponível em: <<http://www.microsoft.com>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Windows**. 2016. Disponível em: <<https://www.microsoft.com/en-us/windows/>>. Acesso em: 11 set. 2016.

ORACLE. **Java**. 2016. Disponível em: <<http://java.com>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **NETBeans**. 2016. Disponível em: <<https://netbeans.org>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Oracle WebLogic Server**. 2016. Disponível em: <<http://www.oracle.com/technetwork/pt/middleware/weblogic/downloads/index.html>>. Acesso em: 11 set. 2016.

PARAISO, Emerson Cabrera; BARTHÈS, Jean-Paul A. A Voice-Enabled Assistant in a Multi-agent System for e-Government Services. In: RAMOS, Felix F.; ROSILLO, Victor Lrios; UNGER, Herwig. Advanced Distributed Systems. In: INTERNATIONAL SCHOOL AND SYMPOSIUM, ISSADS 2005, 5., Guadalajara, Mexico, jan.24-28, 2005. **Proceedings...** Revised Selected Papers. Mexico:

Springer Berlin Heidelberg, 2005. p. 495-503. (3563). Online ISBN: 978-3-540-31674-9, DOI: 10.1007/11533962\_45. Disponível em: <[http://dx.doi.org/10.1007/11533962\\_45](http://dx.doi.org/10.1007/11533962_45)>. Acesso em: 11 set. 2016.

PYTHON SOFTWARE FOUNDATION. **Python**. 2016. Disponível em: <<https://www.python.org/downloads/>>. Acesso em: 11 set. 2016.

RONACHER, Armin. **Flask: web development, one drop at a time**. 2016. Disponível em: <<http://flask.pocoo.org>>. Acesso em: 11 set. 2016.

SILVA, Paulo Caetano da. **Análise multidimensional de dados XML baseados em links: modelos e linguagens**. 2010. 137 f. Tese (Doutorado) - Curso de Pós-graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2010.

SYN. **Syn Engine 2.0**. 2015. Disponível em: <<http://syn.co.in/Syn-Engine.aspx>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Synthetic Intelligence Network**. 2015. Disponível em: <<http://syn.co.in>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **Syn Virtual Assistant**. 2015. Disponível em: <<http://syn.co.in/Syn-Virtual-Assistant.aspx>>. Acesso em: 11 set. 2016.

THE ECLIPSE FOUNDATION. **Eclipse desktops & web IDEs**. 2016. Disponível em: <<https://eclipse.org/ide>>. Acesso em: 11 set. 2016.

U.S. SECURITIES AND EXCHANGE COMMISSION. **Office of Structured Disclosure**. 2016. Disponível em: <<https://www.sec.gov/structureddata>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **SEC.gov Home**. 2016. Disponível em: <<http://www.sec.gov>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **SEC.gov EDGAR Search Tools**. Disponível em: <<https://www.sec.gov/edgar.shtml>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **XBRL SEC**. Disponível em: <<http://www.xbrl.sec.gov>>. Acesso em 31 ago. 2016.

W3C. **JSpeech Grammar Format**, Jun. 2000. Disponível em <<https://www.w3.org/TR/jsgf/>>. Acesso em: 11 set. 2016.

WOLFRAM. **Wolfram**. 2016. Disponível em: <<https://www.wolfram.com/index.php>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **WolframAlpha: computational knowledge engine**. 2016. Disponível em: <<https://www.wolframalpha.com>>. Acesso em: 11 set. 2016.

XBRL INTERNATIONAL. **XBRL**. 2016. Disponível em: <<https://www.xbrl.org>>. Acesso em: 11 set. 2016.

\_\_\_\_\_. **The Standard XBRL**. 2016. Disponível em: <<https://www.xbrl.org/the-standard/>>. Acesso em 11 set. 2016.

XBRL US. **2016 US GAAP Financial Reporting Taxonomy**. 2016. Disponível em: <<https://xbrl.us/xbrl-taxonomy/2016-us-gaap/>>. Acesso em 11 set. 2016.

\_\_\_\_\_. **Data Analysis Toolkit**. 2016. Disponível em: <<https://xbrl.us/home/use/data-analysis-toolkit/>>. Acesso em 11 set. 2016.

\_\_\_\_\_. **How to Use the Data Analysis Toolkit**. 2016. Disponível em: <<https://xbrl.us/wp-content/uploads/2015/10/HowTo-Use-the-Data-Analysis-Toolkit.pdf>>. Acesso em 11 set. 2016.

\_\_\_\_\_. **XBRL US**: The national consortium for the business reporting standard. 2016. Disponível em: <<https://xbrl.us>>. Acesso em 11 set. 2016.

\_\_\_\_\_. **XBRL US GAAP Taxonomy 1.0**. ago. 2008. Disponível em: <<http://www.xbrl.org/Taxonomy/us/2008/gaap/XBRL-US-GAAP-Taxonomy-1.0-Summary-2008.htm>>. Acesso em 11 set. 2016.

ZAMBIASI, Saulo Popov. **Uma arquitetura de referência para softwares assistentes pessoais baseada na arquitetura orientada a serviços**. 2012. 295 f. Tese (Doutorado) - Curso de Programa de Pós-graduação em Engenharia de Automação e Sistemas, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2012. Disponível em: <[http://www.gsigma.ufsc.br/~popov/aulas/Publicacoes/Zambiasi\\_TeseDoutorado\\_2012.pdf](http://www.gsigma.ufsc.br/~popov/aulas/Publicacoes/Zambiasi_TeseDoutorado_2012.pdf)>. Acesso em: 11 set. 2016.

ZAMBIASI, Saulo Popov; RABELO, Ricardo Jose. A Proposal for Reference Architecture for Personal Assistant Software Based on SOA. **IEEE Latin America Transactions**. [S. l.], p. 1227-1234. 02 fev. 2012. ISSN 1548-0992. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6142466>>. Acesso em: 11 set. 2016.

ZHANG, Jun; YANG, Terry Zhenrong; HAZEN, Timothy J. Large-Scaleword Representation Features For Improved Spoken Language Understanding. IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2015. **Proccedings...** 2015. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7178984>>. Acesso em: 11 set. 2016.

## APÊNDICE A - LISTA DE PERGUNTAS UTILIZADAS NOS TESTES

Ordem	Pergunta
1a.	What is the assets of apple in 2015?
2a.	Show me the Current Assets of Google in 2015?
3a.	What is the Goodwill of petrobras in 2008?
4a.	and 2009?
5a.	and liabilities?
6a.	Inform the Liabilities And Equity Total of Microsoft in 2013?
7a.	What is the Liabilities and Stockholders' Equity of Vale in 2009?
8a.	What is the Current Liabilities of apple in 2015?
9a.	What is the Current Liabilities of apple in 2015?
10a.	What is the Current Liabilities of apple in 2015?
11a.	What is the Current Liabilities of apple in 2015?
12a.	What is the Current Liabilities of apple in 2015?
13a.	Show me the Liabilities Current of Microsoft in 2014?
14a.	in 2014?
15a.	in 2015?
16a.	and the assets?
17a.	What is the assets of bradesco in 2009?
18a.	Say the Current Assets of Twitter in 2009?
19a.	Give me Microsoft's Cash and Cash Equivalentents in 2015?
20a.	What is the Cash and Cash Equivalentents of Google in 2014?

21a.	Give me the Liabilities And Equity Total of petrobras in 2008?
22a.	What is the Liabilities and Stockholders' Equity of Facebook in 2014?
23a.	Give me the Current Assets of petrobras in 2009?
24a.	Give me the Current Assets of petrobras in 2009?
25a.	Give me the Current Assets of petrobras in 2009?
26a.	Give me the Current Assets of petrobras in 2009?
27a.	Give me the Current Assets of petrobras in 2009?
28a.	What is the Liabilities Current of twitter in 2009?
29a.	and 2015?
30a.	Say the assets of a apple in 2011?
31a.	Show me the Current Assets of facebook in 2015?
32a.	and Google?
33a.	and in 2008?
34a.	and the Goodwill?
35a.	and Liabilities in 2015?
36a.	What is the Cash and Cash Equivalentents of gol in 2008?
37a.	Inform is the Liabilities And Equity Total of Google in 2015?
38a.	What is the Liabilities and Stockholders' Equity of Microsoft in 2012?
39a.	Give me the Current Liabilities of twitter in 2009?
40a.	Inform the Current Liabilities of Vale in 2009?
41a.	Show me the Apple's Current Assets in 2015?
42a.	What is the Liabilities Current of Itau in 2008?
43a.	and about petrobras?

## APÊNDICE B – AMOSTRA DE UMA MEDIÇÃO REALIZADA NO NAVEGADOR WEB

Obs.: Essa parte da amostra corresponde ao resultado da submissão em sequência de uma lista de perguntas ao Assistente Financeiro Virtual, feito através de medições realizadas por um javascript no navegador web.

Asked Question: What is the assets of apple in 2015  
 Response: The Assets of APPLE INC Company in 2015 is US\$290,479.00  
 Overall Execution Time: 2455.040000000001 miliseconds.  
 Asked Question: Show me the Current Assets of Google in 2015  
 Response: Sorry! Unavailable Information for me!  
 Overall Execution Time: 1253.6599999999999 miliseconds.

Asked Question: What is the Goodwill of petrobras in 2008  
 Response: The Goodwill of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2008 is US\$118.00  
 Overall Execution Time: 1234.0399999999972 miliseconds.

Asked Question: and 2009  
 Response: The Goodwill of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$139.00  
 Overall Execution Time: 1233.8649999999998 miliseconds.

Asked Question: and liabilities  
 Response: Sorry! Unavailable Information for me!  
 Overall Execution Time: 1389.1299999999974 miliseconds.

Asked Question: Inform the Liabilities And Equity Total of Microsoft in 2013  
 Response: The Liabilities and Stockholders' Equity of MICROSOFT CORPORATION Company in 2013 is US\$142,431.00  
 Overall Execution Time: 1238.7949999999983 miliseconds.  
 Asked Question: What is the Liabilities and Stockholders' Equity of Vale in 2009  
 Response: The Liabilities and Stockholders' Equity of Vale S.A. Company in 2009 is US\$102,279.00  
 Overall Execution Time: 1102 miliseconds.

Asked Question: What is the Current Liabilities of apple in 2015  
 Response: The Current Liabilities of APPLE INC Company in 2015 is US\$80,610.00  
 Overall Execution Time: 1087.4850000000006 miliseconds.

Asked Question: What is the Current Liabilities of apple in 2015  
 Response: The Current Liabilities of APPLE INC Company in 2015 is US\$80,610.00  
 Overall Execution Time: 1092.0900000000038 miliseconds.

Asked Question: What is the Current Liabilities of apple in 2015  
 Response: The Current Liabilities of APPLE INC Company in 2015 is US\$80,610.00  
 Overall Execution Time: 1081.0549999999993 miliseconds.

Asked Question: What is the Current Liabilities of apple in 2015  
 Response: The Current Liabilities of APPLE INC Company in 2015 is US\$80,610.00  
 Overall Execution Time: 1086.9349999999977 miliseconds.

Asked Question: What is the Current Liabilities of apple in 2015  
 Response: The Current Liabilities of APPLE INC Company in 2015 is US\$80,610.00  
 Overall Execution Time: 1276.4899999999998 miliseconds.

Asked Question: Show me the Liabilities Current of Microsoft in 2014  
 Response: The Current Liabilities of MICROSOFT CORPORATION Company in 2014 is US\$45,625.00  
 Overall Execution Time: 1111.5950000000012 miliseconds.

Asked Question: in 2014  
 Response: The Current Liabilities of MICROSOFT CORPORATION Company in 2014 is US\$45,625.00

Overall Execution Time: 1131.8899999999994 milliseconds.

Asked Question: in 2015

Response: The Current Liabilities of MICROSOFT CORPORATION Company in 2015 is US\$49,858.00

Overall Execution Time: 1246.1100000000006 milliseconds.

Asked Question: and the assets

Response: The Assets of MICROSOFT CORPORATION Company in 2015 is US\$176,223.00

Overall Execution Time: 1499.7649999999994 milliseconds.

Asked Question: What is the assets of bradesco in 2009

Response: The Assets of BANK BRADESCO Company in 2009 is R\$ 496.815,00

Overall Execution Time: 1114.1600000000035 milliseconds.

Asked Question: Say me the Current Assets of Twitter in 2009

Response: Sorry! Unavailable Information for me!

Overall Execution Time: 1085.135000000002 milliseconds.

Asked Question: Give me Microsoft's Cash and Cash Equivalents in 2015

Response: Sorry! Unavailable Information for me!

Overall Execution Time: 1108.2299999999996 milliseconds.

Asked Question: What is the Cash and Cash Equivalents of Google in 2014

Response: Sorry! Unavailable Information for me!

Overall Execution Time: 1107.5 milliseconds.

Asked Question: Give me the Liabilities And Equity Total of petrobras in 2008

Response: The Liabilities and Stockholders' Equity of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2008 is US\$125,695,000,000.00

Overall Execution Time: 1103.7799999999988 milliseconds.

Asked Question: What is the Liabilities and Stockholders' Equity of Facebook in 2014

Response: The Liabilities and Stockholders' Equity of FACEBOOK INC Company in 2014 is US\$39,966.00

Overall Execution Time: 1157.8400000000038 milliseconds.

Asked Question: Give me the Current Assets of petrobras in 2009

Response: The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00

Overall Execution Time: 1252.520000000004 milliseconds.

Asked Question: Give me the Current Assets of petrobras in 2009

Response: The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00

Overall Execution Time: 1216.4099999999962 milliseconds.

Asked Question: Give me the Current Assets of petrobras in 2009

Response: The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00

Overall Execution Time: 1257.3499999999985 milliseconds.

Asked Question: Give me the Current Assets of petrobras in 2009

Response: The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00

Overall Execution Time: 1135.2399999999998 milliseconds.

Asked Question: Give me the Current Assets of petrobras in 2009

Response: The Current Assets of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2009 is US\$42,644.00

Overall Execution Time: 1155.4399999999995 milliseconds.

Asked Question: What is the Liabilities Current of twitter in 2009

Response: Sorry! Unavailable Information for me!

Overall Execution Time: 1080.2400000000052 milliseconds.

Asked Question: and 2015

Response: The Current Liabilities of TWITTER, INC. Company in 2015 is US\$506,039.00

Overall Execution Time: 1078.2399999999998 milliseconds.

Asked Question: Say me the assets of apple in 2011  
Response: The Assets of APPLE INC Company in 2011 is US\$116,371.00  
Overall Execution Time: 1268.7649999999994 miliseconds.

Asked Question: Show me the Current Assets of facebook in 2015  
Response: The Current Assets of FACEBOOK INC Company in 2015 is US\$21,652.00  
Overall Execution Time: 1322.875 miliseconds.

Asked Question: and Google  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1257.2449999999953 miliseconds.

Asked Question: and in 2008  
Response: The Current Assets of Google Inc. Company in 2008 is US\$20,178,182.00  
Overall Execution Time: 1205.0149999999994 miliseconds.

Asked Question: and the Goodwill  
Response: The Goodwill of Google Inc. Company in 2008 is US\$4,839,854.00  
Overall Execution Time: 1219.590000000011 miliseconds.

Asked Question: and Liabilities in 2015  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1279.770000000004 miliseconds.

Asked Question: What is the Cash and Cash Equivalents of gol in 2008  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1207.675000000003 miliseconds.

Asked Question: Inform is the Liabilities And Equity Total of Google in 2015  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1119.7599999999948 miliseconds.

Asked Question: What is the Liabilities and Stockholders' Equity of Microsoft in 2012  
Response: The Liabilities and Stockholders' Equity of MICROSOFT CORPORATION Company in 2012 is US\$121,271.00  
Overall Execution Time: 1103.2000000000116 miliseconds.

Asked Question: Give me the Current Liabilities of twitter in 2009  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1099.804999999993 miliseconds.

Asked Question: Inform the Current Liabilities of Vale in 2009  
Response: The Current Liabilities of Vale S.A. Company in 2009 is US\$9,181.00  
Overall Execution Time: 1138.3399999999965 miliseconds.

Asked Question: Show me the Apple's Current Assets in 2015  
Response: The Current Assets of APPLE INC Company in 2015 is US\$89,378.00  
Overall Execution Time: 1104.6000000000058 miliseconds.

Asked Question: What is the Liabilities Current of Itau in 2008  
Response: Sorry! Unavailable Information for me!  
Overall Execution Time: 1099.3199999999924 miliseconds.

Asked Question: and about petrobras  
Response: The Current Liabilities of PETROBRAS - PETROLEO BRASILEIRO SA Company in 2008 is US\$24,756,000,000.00  
Overall Execution Time: 1083.4549999999872 miliseconds



## ANEXO A – XML DE RESPOSTA DE UM SERVIÇO DE DADOS EM XBRL

Obs.: Resposta em formato XML do serviço da XBRL-US sobre um dado financeiro da empresa Apple.

```
<dataRequest date="2016-07-28T21:56:30-0400">
  <fact>
    <entity>APPLE INC</entity>
    <entityCode>0000320193</entityCode>
    <accessionID>163863</accessionID>
    <filingAccession>0001628280-16-017809</filingAccession>
    <elementName>CashAndCashEquivalentsAtCarryingValue</elementName>
    <namespace>http://fasb.org/us-gaap/2015-01-31</namespace>
    <extensionflag>N</extensionflag>
    <axis/>
    <axisNamespace/>
    <member/>
    <memberNamespace/>
    <units>USD</units>
    <amount>13844000000</amount>
    <decimals>-6</decimals>
    <fact>13844000000</fact>
    <period>Y</period>
    <year>2014</year>
    <periodStart/><periodEnd/>
    <periodInstant>2014-09-28</periodInstant>
    <filingDate>2016-07-27</filingDate>
    <aligned/>
    <factID>129384240</factID>
    <secURL>
      http://www.sec.gov/Archives/edgar/data/320193/000162828016017809/0001628280-16-017809-
      index.htm
    </secURL>
    <dimensions/>
    <dimensionCount>0</dimensionCount>
    <url>
      http://csuite.xbrl.us/php/dispatch.php?Task=htmlExport&FactID=129384240
    </url>
  </fact>
  <count>1</count>
</dataRequest>
```