

**Pontifícia Universidade Católica do Rio de Janeiro**

**Assistente Virtual para Indicações em Viagens**

**Lucas Hardman Gomes Campos França**

**Primeiro relatório de Projeto Final de Graduação**

**Centro Técnico Científico - CTC**

**Departamento de Informática**

Curso de Graduação em Ciência da Computação

Rio de Janeiro, Julho de 2019



**Lucas Hardman Gomes Campos França**

**Assistente Virtual para Indicações em Viagens**

Relatório de Projeto Final, apresentado ao curso de Ciência da Computação da PUC-Rio como requisito parcial para a obtenção de título de Bacharel em Ciência da Computação.

Orientador: Edmundo Torreão

Rio de Janeiro,

Julho de 2019.

**Índice**

[1. Proposta e Objetivo do Projeto 4](#_Toc10999816)

[2. Especificação de Requisitos 5](#_Toc10999817)

[3. Viabilidade Técnica e Arquitetura de Software 6](#_Toc10999818)

[4. Descrição e Utilização das APIs 9](#_Toc10999819)

[**4.1.** **Introdução ao Watson** 9](#_Toc10999820)

[**4.2.** **Watson Language Translator** 10](#_Toc10999821)

[**4.3.** **Watson Personality Insights** 12](#_Toc10999822)

[**4.3.1.** **Descrição dos indicadores de personalidade** 12](#_Toc10999823)

[**4.3.2.** **Estrutura do JSON** 18](#_Toc10999824)

[**4.3.3.** **Interpretando os indicadores de personalidade** 24](#_Toc10999825)

[**4.4.** **Facebook** 26](#_Toc10999826)

[**4.5.** **Firebase** 26](#_Toc10999827)

[5. Diagrama de Classes 27](#_Toc10999828)

[6. Casos de Uso e Especificação dos Testes a Serem Realizados 35](#_Toc10999829)

[6.1. Casos de uso e casos de teste 35](#_Toc10999830)

[6.2. Especificações dos perfis de teste 35](#_Toc10999831)

[7. Base de dados 37](#_Toc10999832)

[8. Comentários sobre a Implementação 39](#_Toc10999833)

[9. Conclusão 41](#_Toc10999834)

[**9.1.** **Oportunidades para Trabalhos Futuros** 41](#_Toc10999835)

[10. Referências Bibliográficas 42](#_Toc10999836)

1. **Proposta e Objetivo do Projeto**

Este projeto tem como objetivo o desenvolvimento de software através, construindo um aplicativo iOS capaz de sugerir programas de entretenimento, passeios e eventos em viagens de acordo com a personalidade do usuário. Para que isto seja possível, serão utilizadas diversas APIs e aprender a utilizá-las e integrá-las faz parte do objetivo do projeto.

A proposta principal do aplicativo é fazer sugestões para viagens de acordo com a personalidade do usuário. Para que isto seja possível é utilizada uma API da IBM chamada Watson Personality Insights, que se utiliza de técnicas de análise linguística para inferir características intrínsecas da personalidade dos indivíduos com base em comunicações digitais, como e-mails, mensagens de texto, tweets e postagens de fóruns ou redes sociais. Desta forma, para acessar o aplicativo, a API exige que o usuário faça login pelo Facebook e repasse suas postagens para o Personality Insights.

Em paralelo com o aprendizado sobre as APIs utilizadas, está o aprendizado sobre a linguagem de programação Swift e sobre as características do software, tais como interface e usabilidade.

O usuário poderá navegar pelo aplicativo e fazer pesquisas, selecionando a cidade que deseja consultar e podendo utilizar filtros de categorias de sugestão, como programas de entretenimento noturnos, ao ar livre ou artísticos. O software responderá à pesquisa do usuário com uma lista de sugestões, onde cada uma delas pode ser clicada, causando um redirecionamento para uma página que detalha as informações.

1. **Especificação de Requisitos**

* O aplicativo deve exigir login pelo Facebook antes de dar acesso ao seu conteúdo.
* O aplicativo deve aceitar perfis do Facebook com postagens em português.
* O aplicativo deve utilizar as postagens do Facebook para criar um perfil de personalidade para o usuário.
* O aplicativo deve avisar quando a quantidade de postagens não for o suficiente para a criação de um perfil de personalidade.
* O aplicativo deve ter acesso à uma base de dados com perfis de programas de entretenimento, passeios e eventos para serem oferecidas como sugestões.
* As sugestões devem ter perfis com os mesmos indicadores de personalidade do usuário.
* O perfil de uma sugestão deve ser feito a partir de um conjunto de análises positivas disponíveis no site de críticas TripAdvisor.
* O aplicativo deve ter uma janela específica para pesquisa, onde o usuário deve fornecer uma cidade de destino e escolher se quer utilizar filtros de busca.
* Toda sugestão apresentada para o usuário deve ter um indicador de porcentagem que represente a compatibilidade entre o seu perfil e o perfil do usuário.
* As sugestões devem ser apresentadas ao usuário através de uma lista ordenada decrescentemente por percentual de compatibilidade com o usuário.
* A lista de sugestões deve exibir, para cada sugesão, o nome, a categoria, o início da descrição, uma imagem e uma barra que demonstre visualmente a compatibilidade com o usuário.
* O aplicativo deve possibilitar o usuário marcar sugestões como favorita.
* O aplicativo deve ter uma janela específica para consultar as sugestões marcadas como favoritas.

Não funcionais:

* O aplicativo só pode ser utilizado com conexão à internet.
* Todos os dados sobre o usuário devem estar persistidos em uma base de dados.

1. **Viabilidade Técnica e Arquitetura de Software**

O projeto será implementado para smartphones iOS e utilizará recursos do IBM Watson, Facebook e Firebase. A seguir temos algumas informações sobre as versões dos recursos utilizados.

|  |  |
| --- | --- |
| Linguagem do aplicativo | Swift 4.2 |
| Sistema Operacional do aplicativo | iOS 12.2 |
| Sistema Operacional do ambiente de desenvolvimento | MacOS High Sierra 10.13.6 |
| IDE | Xcode 10.1 |
| Integração com o Watson | https://github.com/watson-developer-cloud/swift-sdk |
| Integração com o Facebook | https://github.com/facebook/facebook-sdk-swift |

Este aplicativo é uma solução de sistemas de recomendação e é baseado em uma publicação da IBM, Sugestões de entretenimento usando computação cognitiva [1]. A publicação sugere uma aplicação de sugestão de filmes de acordo com o perfil do usuário criado pelo Watson Personality Insights. O perfil do usuário é comparado com o perfil do filme, que possui os mesmos indicadores, e caso o perfil seja similar, o filme é sugerido.

Mas como são criados os perfis dos filmes? É escolhido um site de críticas e avaliações, neste caso o IMDb, para que um webcrawler atue buscando críticas positivas (dando nota acima de 7 ao filme, por exemplo) e envie uma lista com essas críticas para o Watson Personality Insights criar um perfil para o filme. Ou seja, o perfil do filme é um perfil criado a partir do público que consome aquele conteúdo.

O foco deste projeto é apenas o aplicativo de sugestões para viagens que será utilizado pelo usuário final. Então a criação dos perfis dos eventos, estabelecimentos e pontos turísticos sugeridos será feita de uma outra forma. Será desenvolvido um outro aplicativo de apoio, também descrito neste relatório, para ler um perfil do Facebook das possíveis sugestões, criar um perfil Watson Personality Insights e enviar para o Firebase. Este perfil do Facebook será criado ‘à mão’ de forma que cada postagem seja uma crítica com nota maior ou igual a 7, retirada de sites de críticas e análises dos consumidores.

A partir deste momento, vamos tratar como “sugestão turística” tudo aquilo que a aplicação pode sugerir para o usuário, como estabelecimentos, eventos e pontos turísticos. Existem 3 principais processos para este projeto.

1 – Adicionar sugestões turísticas para os usuários no Firebase:

A screenshot of a cell phone

Description automatically generated

Diagrama de atividades – Adição de sugestão turística

* Foram criados usuários de teste no Facebook para cada sugestão turística. Para cada um desses usuários, foram adicionadas diversas postagens com avaliações com notas máximas retiradas do site TripAdvisor manualmente.
* Ao entrar no aplicativo, o administrador encontra um formulário pedindo nome da sugestão turística, a cidade em que ela se encontra, a sua categoria, o link para o seu site oficial, um link para uma imagem e um texto descritivo.
* Após preencher o formulário, o administrador clica no botão enviar e então o aplicativo pede para fazer login no facebook. Devem ser preenchidas credenciais referentes ao usuário de teste criado para aquela sugestão turística.
* Efetuado o login, o aplicativo lê as postagens do perfil do usuário de teste, envia para o Watson Language Translator traduzir para inglês e depois envia o resultado para o Watson Personality Insights gerar um perfil de personalidade para esta sugestão turística.
* Com o formulário preenchido e o perfil de personalidade gerado com sucesso, o aplicativo realiza a persistencia das informações obtidas para o Firebase.
* Após as informações serem enviadas para o Firebase, o administrador deve apertar o botão “Concluir”, para fazer logout do Facebook e retornar ao formulário.

2 – O usuário faz login no aplicativo

A close up of a piece of paper

Description automatically generated

Diagrama de atividades – Login no aplicativo

ESCREVER EXPLICACAO

3 – O usuário faz uma consulta no aplicativo

A screenshot of a cell phone

Description automatically generated

Diagrama de atividades – Efetuar consulta

ESCREVER EXPLICACAO

Colocar as capituras de tela aqui.

1. **Descrição e Utilização das APIs**

Neste projeto são utilizadas as seguintes APIs: Watson Language Translator, Watson Personality Insights, Facebook e Firebase. Nesta sessão está descrita a relação entre cada uma dessas APIs e para cada uma delas temos uma explicação de como ela é aplicada no projeto.

ESCREVER ALGUMA COISA AQUI

* 1. **Introdução ao Watson**

Inicialmente criado pela IBM em 2011 para participar do programa televisivo americano Jeopardy, o Watson é um sistema de computação cognitiva. Desta forma, ele utiliza técnicas de “machine learning” para aprender com experiências anteriores e aplica estes conhecimentos em futuras tomadas de decisão.

O Watson veio para tentar evoluir na solução de um dos grandes problemas da computação: entender e interpretar a linguagem natural. Ele é capaz de entender dados não-estruturados, que representam 80% dos dados encontrados virtualmente em 2017. Estes dados são normalmente informações produzidas por “seres humanos” para “seres humanos”, incluindo textos como artigos, pesquisas e postagens. Ou seja, temos um cenário de um programa computacional tentando interpretar informações com regras como gramática, contexto e cultura, e podem ser ambíguas, implícitas, complexas.

A IBM já oferece diversos serviços do Watson que podem ser utilizados por qualquer pessoa com conhecimentos em computação e programação. Dentre eles temos o Conversation e o Personality Insights. O primeiro é um serviço que torna capaz uma conversa por chat entre um usuário “falante” de linguagem natural e um robô que utiliza a inteligência do Watson. O robô deve entender a linguagem natural e responder da forma “mais humana possível”. Já o segundo serviço lê e interpreta os dados retirados de redes sociais como Facebook e Twitter para identificar características de personalidade, necessidades, gostos, valores, preferências e hábitos. Um outro serviço do Watson é o Language Translator, que permite a tradução de texto entre diferentes línguas através da computação cognitiva, ou seja, a tradução não é apenas literária, mas também leva em conta termos, gírias e combinações de palavras, cujo significado foi aprendido por machine learning.

Já podemos contar com aplicações do Watson em diversas áreas, como por exemplo na medicina, onde ele é capaz de recomendar terapias contra o câncer a partir do cruzamento da literatura científica com dados clínicos e genéticos dos pacientes. Além dessa área, educação, finanças e “internet das coisas” também são focos da IBM para o Watson.

Dessa forma, o Watson é capaz de ser aplicado em diversos setores, entre eles o turismo. Os negócios desta indústria são normalmente diretos e objetivos, como montagem de pacotes de viagens, reservas de hotéis, aluguéis de carros, compras de passagens aéreas e compra de ingressos. O uso da computação cognitiva pode trazer para estes negócios um entendimento sobre as características e necessidades do cliente e com isso oferecer um serviço mais personalizado para cada cliente.

* 1. **Watson Language Translator**

Um dos primeiros problemas encontrados no estudo das APIs para este projeto, foi que o Watson Personality Insights não aceita textos de entrada na língua portuguesa. A primeira solução pensada foi fazer o programa totalmente em inglês, o que seria uma opção razoável apesar de dificultar os testes. Uma outra solução mais completa, seria fazer o projeto em português e utilizar uma ferramenta de tradução para traduzir as postagens do Facebook do usuário para inglês antes de repassá-las para o Watson Personality Insights. A opção escolhida para ser essa ferramenta de tradução foi o Watson Language Translator, pois já está incluído no pacote de serviços que este projeto utiliza e tem o poder de fazer uma tradução mais precisa como veremos mais para frente.

O Watson Language Translator é uma ferramenta poderosa de tradução que utiliza técnicas de deep learning para melhorar a velocidade e a precisão da tradução, o que ajuda a identifcar o signficado de termos, gírias e combinações de palavras. Todos os pares de idiomas utilizam a tradução da rede neural do Watson. O serviço ainda é capaz de identificar o idioma de origem do texto, o que é de grande ajuda para este projeto pois o aplicativo poderar ler um perfil do Facebook independente da língua escrita nas postagens.

O serviço garante a preservação da privacidade dos dados e funciona com os seguintes idiomas: Árabe, Catalão, Chinês (Simplificado e Tradicional), Tcheco, Dinamarquês, Holandês, Inglês, Finlandês, Francês, Alemão, Hindi, Húngaro, Italiano, Japonês, Coreano, Norueguês, Polonês, Português (Brasil), Russo, Espanhol, Espanhol, Sueco e Turco. Existe também uma característica interessante, que não possui aplicação neste projeto, que possibilita a tradução de documentos preservando a formatação do mesmo e pode ser utilizado em arquivos Microsoft Office, Open Office, PDF, HTML, JSON, TXT e XML.

Aplicação do serviço no projeto:

A partir do momento em que a API do Facebook enviar para o aplicativo as postagens do usuário, o Language Translator irá traduzir cada postagem separadamente antes do aplicativo repassar as postagens para o JSON que será lido pelo Personality Insights.

* 1. **Watson Personality Insights**
     1. **Descrição dos indicadores de personalidade**
* **Big Five:** são características de personalidade representadas pelo modelo descrevendo como uma pessoa se envolve com o mundo. Elas possuem sub características (traços) que são chamadas de “facet”.
* **Needs:** indica quais aspectos de um produto podem combinar com uma pessoa.
* **Values:** indica fatores de motivação que influenciam a tomada de decisão de uma pessoa.
* **Consumption Preferences:** baseado nos modelos de características de personalidade Big Five, Needs e Values, o Watson é capaz de produzir um relatório sobre as preferências de consumo do usuário a partir dos seus dados de entrada.

1. **Big Five** – **Agreeableness:** mede a tendência de uma pessoa a ser compassiva e compreensiva com outras pessoas. Em Português, seria agradabilidade.
   1. **Altruism / Altruistic:** ajudar o próximo está mais perto de ser uma realização pessoal do que ser um sacrifício. Português: altruísmo.
   2. **Cooperation / Accommodating / Compilance:** não gosta de confronto. Estão dispostos a se comprometer e abrir mão de suas necessidades para agradar o próximo. Português: cooperação.
   3. **Modesty / Modest:** são despretensiosos e humildes. No entanto não necessariamente têm autoconfiança ou autoestima. Português: modesto.
   4. **Morality / Uncompromising / Sincerity:** não são pretensiosos e manipuladores. Por outro lado, são francos e genuínos. Português: moralidade / sinceridade.
   5. **Sympathy / Empathetic:** são carinhosos e compassivos. Português: simpáticos
   6. **Trust:** assumem que a maior parte das pessoas são justas, honestas e possuem boas intenções. Normalmente estão dispostas a perdoar e esquecer.
2. **Big Five** – **Conscientiousness:** mede a tendência de uma pessoa a agir de forma planejada e organizada. Português: conscienciosidade.
   1. **Achievement striving / Driven:** se esforça para alcançar a excelência. A sua busca por ser reconhecido como caso de sucesso o mantém focado em trabalhar para alcançar seus objetivos. Português: esforço para alcançar seu objetivo.
   2. **Cautiousness / Deliberate:** pensam cuidadosamente sobre as possibilidades e consequências de suas ações antes de tomar alguma decisão. Português: cauteloso.
   3. **Dutiful / Sense of responsability:** possuem um grande senso de responsabilidade e dever. Português: obediente / atencioso.
   4. **Orderliness / Organized:** são bem organizados. Português: organizado.
   5. **Self-discipline / Persistent:** possuem autodisciplina e força de vontade para persistir em tarefas difíceis ou desagradáveis até serem concluídas. Português: autodisciplina / persistência.
   6. **Self-efficacy / Sense of competence:** são confiantes na sua habilidade de finalizar tarefas. Português: senso de competência.
3. **Big Five** – **Extraversion:** mede a tendência de uma pessoa a buscar estímulo na companhia de outras pessoas. Português: extrovertido.
   1. **Activity level / Energetic:** se movimentam e fazem atividades de maneira rápida, energética e vigorosa. Normalmente estão envolvidos em várias tarefas. Português: ativo / energético.
   2. **Assertiveness:** gostam de assumir o controle de direcionar as tarefas de outras pessoas. Tendem a ser líderes de grupos. Português: assertividade.
   3. **Cheerfulness / Positive emotions:** frequentemente experienciam sentimentos positivos, como felicidade, prazer e otimismo. Português: alegre.
   4. **Excitement-seeking:** normalmente precisam de estímulos elevados para não ficarem entediados. Português: intensos.
   5. **Friendliness / Outgoing / Warmth:** normalmente gostam de outras pessoas e demonstram sentimentos positivos publicamente. Português: caloroso.
   6. **Gargariousness / Sociable:** acham que a companhia dos outros é agradavelmente estimulante e gratificante. Tendem a gostar da emoção das multidões. Português: sociável.
4. **Big Five** – **Emotional Range:** também conhecida como neuroticismo, é a medida em que uma pessoa é sensível às emoções negativas do ambiente.
   1. **Anger / Fiery:** tendência de sentir raiva. Português: raiva.
   2. **Anxiety:** frequentemente sente que algo desagradável, perigoso ou ameaçador está para acontecer. Português: ansiedade.
   3. **Depression / Melancholy:** tendem a reagir mais prontamente aos altos e baixos da vida. Português: depressão / melancolia.
   4. **Immoderation / Self-indulgence:** tendem a ser orientados para prazeres e recompensas de curto prazo em vez de consequências de longo prazo. Português: imediatismo.
   5. **Self-consciousness:** são sensíveis ao que os outros pensam sobre eles. Estão preocupados com rejeições e acontecimentos vergonhosos. Sentem-se envergonhados e inconfortáveis perto de outras pessoas com frequência. Português: envergonhados.
   6. **Vulnerability / Sensitivity to stress:** não sabem lidar com estresse. Podem se sentir confusos, sozinhos e em pânico quando estão sob pressão ou quando estão diante a uma situação emergencial.
5. **Big Five** – **Openess:** mede a tendência de uma pessoa estar aberta a novas experiências. Português: capacidade receptiva.
   1. **Adventurousness:** estão sempre procurando experimentar novas atividades. Para eles a familiaridade e a rotina devem ser evitadas. Português: aventureiro.
   2. **Artistic Interests:** amam a beleza, tanto na arte quanto na natureza. São facilmente envolvidos em eventos naturais e artísticos. Português: interesses artísticos.
   3. **Emotionality / Depth of emotions:** possuem consciência de seus próprios sentimentos. Português: emotividade.
   4. **Imagination:** usam a fantasia como um caminho para criar um mundo interior mais rico e interessante do que o mundo real. Português: imaginação.
   5. **Intellect:** são intelectualmente curiosos e tendem a pensar em símbolos e abstrações. Português: intelectual.
   6. **Liberalism / Tolerance for diversity:** estão de prontidão para desafiar a autoridade, a convenção e os valores tradicionais. Português: liberalismo / tolerância a diversidade.
6. **Needs:**
   1. **Excitement:** querem sair e viver a vida, ter emoções animadas e se divertir. Português: excitação.
   2. **Harmony:** apreciam as outras pessoas, seus pontos de vista e seus sentimentos. Português: harmonia.
   3. **Curiosity:** tem a curiosidade de conhecer coisas novas, aprender e crescer. Português: curiosidade.
   4. **Ideal:** almejam a perfeição e a melhoria da comunidade. Português: idealista.
   5. **Closeness:** preferem estar conectadas à família e estabelecer um lar. Português: valorizam a família.
   6. **Self-expression:** gostam de aprender mais sobre si mesmos e se autoafirmar. Português: auto-expressão.
   7. **Liberty:** possuem gostos por moda e novidades. Precisam de espaço para estar livres. Português: liberdade.
   8. **Love:** gostam de contato social, tanto em grupo como em casal. Se sentem envolvidos em campanhas de reunir as pessoas. Português: amável.
   9. **Practicality:** desejam ter o trabalho feito/pronto. Buscam eficiência e habilidade, que podem incluir experiência e expressão física. Português: praticidade.
   10. **Stability:** buscam por estabilidade. Preferem o que foi provado e testado. Português: estabilidade.
   11. **Challenge:** possuem ânsia em alcançar os objetivos e serem bem-sucedidos. Português: desafio.
   12. **Structure:** tem a necessidade de que as coisas precisam estar organizadas e sob controle. Português: não toleram a inexistência de um padrão.
7. **Values:**
   1. **Self-transcedence / Helping others:** demonstram preocupação com o bem-estar e os interesses dos outros. Português: auto transcendência.
   2. **Conservation / Tradition:** enfatizam a autorrestrição e ordem. Resistem às mudanças. Português: conservador.
   3. **Hedonism / Taking pleasure in life:** procuram prazer e gratificação para si mesmos. Português: hedonismo.
   4. **Self-enhancement / Achieving success:** procuram sucesso e melhorias para si mesmos. Português: auto-aprimoramento.
   5. **Open to change / Excitement:** estão abertos a novas experiências e enfatizam ações independentes. Português: aberto a mudanças.
8. **Consumption Preferences:**
9. **Compras:**
   1. **consumption\_preferences\_automobile\_ownership\_cost:** consideram o gasto total da aquisição de automóveis antes de concluir uma escolha.
   2. **consumption\_preferences\_automobile\_safety:** consideram os fatores de segurança para comprar automóveis.
   3. **consumption\_preferences\_clothes\_quality:** preferem qualidade ao comprar roupas.
   4. **consumption\_preferences\_clothes\_style:** preferem estilo e moda ao comprar roupas.
   5. **consumption\_preferences\_clothes\_comfort:** preferem conforto ao comprar roupas.
   6. **consumption\_preferences\_influence\_brand\_name:** são influenciados pela marca do produto ao fazer compras.
   7. **consumption\_preferences\_influence\_utility:** são influenciados pela utilidade do produto antes de efetuar a compra.
   8. **consumption\_preferences\_influence\_online\_ads:** são influenciados por propagandas online antes de efetuar compras.
   9. **consumption\_preferences\_influence\_social\_media:** são influenciados por redes sociais antes de efetuar compras.
   10. **consumption\_preferences\_influence\_family\_members:** são influenciados por membros da família antes efetuar compras.
   11. **consumption\_preferences\_spur\_of\_moment:** podem efetuar compras por impulso.
   12. **consumption\_preferences\_credit\_card\_payment:** preferem utilizar cartão de crédito para efetuar compras.
10. **Filmes:**
11. **consumption\_preferences\_movie\_romance:** tendem a preferir filmes de romance.
12. **consumption\_preferences\_movie\_adventure:** tendem a preferir filmes de aventura.
13. **consumption\_preferences\_movie\_horror:** tendem a preferir filmes de horror.
14. **consumption\_preferences\_movie\_musical:** tendem a preferir musicais.
15. **consumption\_preferences\_movie\_historical:** tendem a preferir filmes históricos.
16. **consumption\_preferences\_movie\_science\_fiction:** tendem a preferir filmes de ficção científica.
17. **consumption\_preferences\_movie\_war:** tendem a preferir filmes de guerra.
18. **consumption\_preferences\_movie\_drama:** tendem a preferir filmes de drama.
19. **consumption\_preferences\_movie\_action:** tendem a preferir filmes de ação.
20. **consumption\_preferences\_movie\_documentary:** tendem a preferir documentários.
21. **Música:**
22. **consumption\_preferences\_music\_rap:** tendem a gostar do estilo Rap.
23. **consumption\_preferences\_music\_country:** tendem a gostar do estilo Country.
24. **consumption\_preferences\_music\_r\_b:** tendem a gostar do estilo R&B.
25. **consumption\_preferences\_music\_hip\_hop:** tendem a gostar do estilo Hip Hop.
26. **consumption\_preferences\_music\_live\_event:** tendem a gostar de shows e eventos ao vivo.
27. **consumption\_preferences\_music\_playing:** possivelmente possuem experiências em tocar algum instrumento.
28. **consumption\_preferences\_music\_latin:** tendem a gostar de música latina
29. **consumption\_preferences\_music\_rock:** tendem a gostar do estilo Rock.
30. **consumption\_preferences\_music\_classical:** tendem a gostar de música clássica.
31. **Leitura e aprendizado:**
32. **consumption\_preferences\_read\_frequency:** tendem a ler com frequência.
33. **consumption\_preferences\_books\_entertainment\_magazines:** tendem a ler revistas de entretenimento.
34. **consumption\_preferences\_books\_non\_fiction:** tendem a ler livros que não são de ficção.
35. **consumption\_preferences\_books\_financial\_investing:** tendem a ler livros sobre investimento financeiro.
36. **consumption\_preferences\_books\_autobiographies:** tendem a ler autobiografias.
37. **Saúde e atividade física:**
38. **consumption\_preferences\_eat\_out:** tendem a comer em restaurantes ou fora de casa com frequência.
39. **consumption\_preferences\_gym\_membership:** tendem a estar inscritos em alguma academia ou pratica de esporte.
40. **consumption\_preferences\_outdoor:** tendem a gostar de atividades ao ar livre.
41. **Empreendedorismo:**
42. **consumption\_preferences\_start\_business:** podem considerar começar um novo negócio nos próximos anos.
43. **Meio ambiente:**
44. **consumption\_preferences\_concerned\_environment:** demonstram interesse e preocupação sobre questões ecológicas.
45. **Voluntariado:**
46. **consumption\_preferences\_volunteer:** tendem a se voluntariar para ajudar em causas sociais.
    * 1. **Estrutura do JSON**

O Watson Personality Insights, ao ler um texto ou conjunto de textos, retorna um objeto JSON contendo os indicadores de personalidade. Este objeto se chama *Profile* e possui as seguintes propriedades:

* **word\_coun**t**:** inteiro que representa a quantidade de palavras lidas do arquivo de entrada. Este inteiro pode conter um valor menor do que o valor real de palavras do arquivo, caso este arquivo seja muito grande.
* **processed\_language:** string que representa o nome da linguagem natural que o serviço utilizou para processar o arquivo de entrada. Por exemplo: en (inglês) ou es (espanhol).
* **personality:** array recursivo de objetos Trait que descrevem cada característica do usuário (Big Five dimensions e facets), obtida do arquivo de entrada.
* **needs:** array de objetos Trait que descreve cada necessidade do usuário (Needs) obtida do arquivo de entrada.
* **values:** objeto de Trait que descreve cada valor do usuário (Values) obtido do arquivo de entrada.
* **behavior:** array de objetos Behavior que descreve a distribuição do comportamento do usuário, através da informação gerada distribuída pelos dias das semanas e horas dos dias. Este campo só é retornado caso o arquivo JSON de entrada possua um campo de horário de data.
* **consumption\_preferences:** array de objetos ConsumptionPreferencesCategory que fornecem resultados para cada categoria de preferência de consumo. O retorno deste campo é opcional e para acontecer o parâmetro consumption\_preferences deve ser setado como true.
* **warnings:** array de objetos Warning que geram mensagens de alerta associadas ao arquivo de entrada. Caso não haja alertas o array é vazio.

Exemplo:

|  |
| --- |
| {  "word\_count": 15223,  "processed\_language": "en",  "personality": [  . . .  ],  "needs": [  . . .  ],  "values": [  . . .  ],  "behavior": [  . . .  ],  "consumption\_preferences": [  . . .  ],  "warnings": []  } |

O objeto Profile sempre inclui arrays de objetos Trait que representam Personality, Needs e Values. Para Needs e Values, estes arrays possuem apenas um nível, contendo em cada posição informações sobre uma necessidade ou valor. Já para o Personality, o array é duplo e representa as características Big Five. Cada característica Big Five possui um campo “children” que representa um novo array de objetos Trait, e em cada posição possui uma característica específica de cada Big Five. Cada posição do array de objetos Trait possui os seguintes campos:

* **trait\_id**: id único, em forma de string, para cada característica. Este id possui os formatos abaixo, trocando “characteristic” pelo nome da característica.
  + **big5\_characteristic:** dimensões Big Five. Ex: big5\_openess.
  + **facet\_characteristic:** características da dimensão Big Five. Ex: facet\_altruism.
  + **need\_characteristic:** necessidades Need. Ex: need\_challenge.
  + **value\_characteristic:** valores Value. Ex: value\_conservation.
* **name**: uma string legível pelo usuário contendo o nome da característica.
* **category**: categoria da característica. Pode ser uma das três opções abaixo:
  + **personality:** para dimensões Big Five ou facets.
  + **needs:** para necessidades.
  + **values:** para valores.
* **percentile:** um número do tipo double que contém a pontuação da característica em percentil.
* **raw\_score:** é um número do tipo double que contem a pontuação bruta da característica. Este campo é opcional, só retorna caso o parâmetro raw\_scores seja true na chamada do método getProfile().
* **siginificant:** é um booleano que sempre é true para inglês, espanhol e japonês. Mas pode ser false para as línguas árabe e coreano, que ainda possuem certa limitação no serviço. Devido a esta limitação os resultados podem não ser confiáveis.
* **children:** é um array de objetos Trait que fornece resultados detalhados para cada característica (facet) de cada dimensão Big Five. Só é retornado quando o trait\_id for uma big5\_characteristic

Exemplo:

|  |
| --- |
| {  . . .  "personality": [  {  "trait\_id": "big5\_openness",  "name": "Openness",  "category": "personality",  "percentile": 0.8011555009553,  "raw\_score": 0.77565404255038,  "significant": true,  "children": [  {  "trait\_id": "facet\_adventurousness",  "name": "Adventurousness",  "category": "personality",  "percentile": 0.89755869047319,  "raw\_score": 0.54990704031219,  "significant": true  },  . . .  ]  },  {  "trait\_id": "big5\_conscientiousness",  . . .  },  {  "trait\_id": "big5\_extraversion",  . . .  },  {  "trait\_id": "big5\_agreeableness",  . . .  },  {  "trait\_id": "big5\_neuroticism",  . . .  }  ],  "needs": [  {  "trait\_id": "need\_challenge",  "name": "Challenge",  "category": "needs",  "percentile": 0.67362332054511,  "raw\_score": 0.75196348037675,  "significant": true  },  . . .  ],  "values": [  {  "trait\_id": "value\_conservation",  "name": "Conservation",  "category": "values",  "percentile": 0.89268222856139,  "raw\_score": 0.72135308187423,  "significant": true  },  . . .  ],  . . .  } |

Caso o JSON de entrada contenha marcadores de tempo (dia e hora), o objeto Profile irá conter um campo com um objeto do tipo Behavior para cada dia da semana e cada hora do dia. Este objeto inclui:

* **trait\_id:** id único, em forma de string, para cada característica. Este id possui os formatos abaixo, trocando “day” pelo dia da semana e “hour” pelo horário do dia.
  + **behavior\_day:** para dias da semana. Ex: behavior\_sunday.
  + **behavior\_hour:** para horas do dia: Ex: behavior\_0100, 1hora da manhã.
* **name:** string legível pelo usuário contendo o dia da semana ou horário do dia. Ex: “Sunday”, ou “0:00 am”
* **category:** string contendo acategoria da característica, sempre será “behavior”
* **percentage:** um número double que representa a percentagem de objetos ContentItems que ocorreram durante certa hora do dia ou dia da semana.

Para o projeto em questão o objeto Behavior não é importante, então não há preocupação em adicionar marcadores de dias e horas no JSON de entrada.

Exemplo:

|  |
| --- |
| {  . . .  "behavior": [  {  "trait\_id": "behavior\_sunday",  "name": "Sunday",  "category": "behavior",  "percentage": 0.21392532795156  },  . . .  ],  . . .  } |

Como vimos anteriormente, ao fazer o requerimento de um Profile, podemos passar como parâmetro true no campo consumption\_preferences, e assim receber um objeto ConsumptionPreferencesCategory que detalha as preferências de consumo do usuário. Este objeto contém:

* **consumption\_preference\_category\_id:** string contendo um id único para cada categoria de preferência de consumo. Ex: “consumption\_preferences\_shopping”
* **name:** string contendo o nome da categoria de forma legível para o usuário.
* **consumption\_preferences:** um array de objetos do tipo ConsumptionPreferences, que contém resultados detalhados de cada preferência de consumo do usuário.

Algumas categorias de preferência de consumo possuem apenas uma preferência, outras possuem várias. Cada preferência é detalhada a partir de um objeto ConsumptionPreferences, que contém os seguintes campos:

* **consumption\_preference\_id:** string contendo um id único para cada preferência de consumo. Ex: "consumption\_preferences\_automobile\_ownership\_cost".
* **name:** string contendo o nome da categoria de forma legível para o usuário. Ex: Likely to be sensitive to ownership cost when buying automobiles".
* **score:** número double que contem a pontuação que indica a probabilidade do usuário preferir tal item.

Exemplo:

|  |
| --- |
| {  . . .  "consumption\_preferences": [  {  "consumption\_preference\_category\_id": "consumption\_preferences\_shopping",  "name": "Purchasing Preferences",  "consumption\_preferences": [  {  "consumption\_preference\_id": "consumption\_preferences\_automobile\_ownership\_cost",  "name": "Likely to be sensitive to ownership cost when buying automobiles",  "score": 0  },  . . .  ]  },  . . .  } |

* + 1. **Interpretando os indicadores de personalidade**

Neste ponto já estudamos o objeto Profile completo e sabemos que ele detalha resultados em objetos do tipo Trait (personalidade: Big Five e facet), Behavior (comportamento) e ConsumptionPreferences (preferências de consumo). Todos estes objetos detalham características da personalidade do usuário com indicadores numéricos do tipo double.

1. **Percentil de características de personalidade (*percentile*):**

Para cada requerimento, o serviço fornece uma pontuação através do atribulo *percentile* para cada característica de personalidade Big Five, Needs e Values. Esta pontuação representa um valor de percentagem (variando de 0 a 1) para cada característica baseado no texto de entrada, e é normalizada através de uma comparação com dados obtidos de uma amostra da população.

Por exemplo, o *percentile* 0.64980796071382 da característica big5\_extraversion indica que o usuário pontuou esta característica mais do 64% população e menos do que 34%.

1. **Valores brutos de características de personalidade (*raw\_score*):**

Os valores brutos *raw\_score* são os mesmos que os *percentile*, porém obtidos sem a normalização com uma amostra da população. Eles podem ser interpretados como resultados obtidos em uma prova de personalidade.

Estes valores são fornecidos para desenvolvedores que querem implementar uma normalização personalizada. Por exemplo, criando uma normalização diferente levando em conta a região e cultura. Por isto estes valores não são retornados por default, e para retornar é necessário passar *true* como parâmetro *raw\_scores* na chamada do método.

1. **Percentagens de características de comportamento: (*percentage*):**

Caso haja indicadores de tempo no JSON de entrada, o serviço retorna uma percentagem para cada característica de comportamento. Isto serve para identificar uma distribuição temporal dos dados analizados, ou seja, o valor indica qual percentagem dos textos de entrada está distribuído em cada hora do dia ou dia da semana.

Por exemplo, a pontuação da característica behavior\_sunday foi 0.4561049445005. Isto significa que 46% dos textos analizados (*ContentItems*) coram criados no domingo.

1. **Pontuação por preferência de consumo (*scores*):**

Conforme interesse, o Personality Insights pode retornar uma análise sobre os interesses de consumo do usuário do texto analizado. Para isto basta especificar como *true* o parâmetro *consumption\_preferences* na hora de fazer o requerimento de um objeto Profile.

Para representar os interesses de consumo, cada objeto do tipo ConsumptionPreferences possui um atributo *score*, que é um indicador numérico double. Este atributo é um indicador de preferência e representa a probabilidade de o usuário preferir tal item.

O atributo *score* pode possuir um dos três valores a seguir:

* **0.0:** Interesse muito baixo em certo item.
* **0.5:** O usuário é neutro quanto ao item.
* **1.0:** Alto nível de interesse no item.
  1. **Facebook**
  2. **Firebase**

1. **Diagrama de Classes**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated**

**A screenshot of a social media post

Description automatically generated A screenshot of a social media post

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a device

Description automatically generated A close up of a device

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

1. **Casos de Uso e Especificação dos Testes a Serem Realizados**
   1. **Casos de uso e casos de teste**
   2. **Especificações dos perfis de teste**

Perfis de usuário:

Perfis de indicações para viagens:

|  |
| --- |
| Nome: Roland Garros |
| Email: perfil\_atvqfuy\_garros@tfbnw.net |
| Cidade: Paris  Categoria: Esportes |
| Link para o site: <https://www.rolandgarros.com/en-us/>  Link para a imagem: ?????????????????????????? |
| Descrição:  O Torneio de Roland Garros (Internationaux de France, French Open ou Aberto da França) é um torneio de tênis realizado em Paris, na França. Tem seu nome em homenagem a Roland Garros, francês pioneiro da aviação.  Com o Australian Open, o Torneio de Wimbledon e o US Open, o torneio de Roland Garros compõe os quatro torneios do Grand Slam de tênis. É disputado em quadra de saibro, em melhor de 5 sets para os homens e 3 sets para as mulheres. Em 1968, o torneio de Roland Garros foi o primeiro torneio do Grand Slam a ser "aberto", permitindo a participação tanto de amadores como de profissionais.  Fonte: <https://pt.wikipedia.org/wiki/Torneio_de_Roland_Garros> |

|  |
| --- |
| Nome: Roland Garros |
| Email: perfil\_atvqfuy\_garros@tfbnw.net |
| Cidade: Paris  Categoria: Esportes |
| Link para o site: <https://www.rolandgarros.com/en-us/>  Link para a imagem: ?????????????????????????? |
| Descrição:  O Torneio de Roland Garros (Internationaux de France, French Open ou Aberto da França) é um torneio de tênis realizado em Paris, na França. Tem seu nome em homenagem a Roland Garros, francês pioneiro da aviação.  Com o Australian Open, o Torneio de Wimbledon e o US Open, o torneio de Roland Garros compõe os quatro torneios do Grand Slam de tênis. É disputado em quadra de saibro, em melhor de 5 sets para os homens e 3 sets para as mulheres. Em 1968, o torneio de Roland Garros foi o primeiro torneio do Grand Slam a ser "aberto", permitindo a participação tanto de amadores como de profissionais.  Fonte: <https://pt.wikipedia.org/wiki/Torneio_de_Roland_Garros> |

1. **Base de dados**

Sugestão turística:

**Id:** número identificador da sugestão turística, retirado do Facebook. Chave primária.

**Nome:** nome da sugestão turística, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator.

**Cidade:** cidade onde a sugestão turística se encontra, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator.

**Categoria:** categoria da sugestão turística, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator. Só pode ser uma das seguintes opções:

* Natureza. *Exemplos: parques nacionais, fazendas, grutas, praias.*
* Esportes. *Exemplos: eventos esportivos, ginásios, estádios, trilhas, pontos de práticas esportivas.*
* Arte. *Exemplos: peças teatrais, concertos musicais.*
* Cultura. *Exemplos: museus, pontos históricos.*
* Tecnologia. *Exemplos: eventos de jogos ou tecnologia, empresas.*
* Vida noturna. *Exemplos: bares, pubs, boates, feiras noturnas, festas.*
* Landmarks. *Exemplos: Torre Eiffel, Cristo Redentor, Estátua da Liberdade, Muralha da China.*
* Gastronomia. *Exemplos: restaurantes, cafés, feiras gastronômicas.*
* Compras. *Exemplos: shopping centers, bazares, feiras.*
* Negócios. *Exemplos: workshops, conferências, eventos de finanças, marketing, etc.*
* Família. *Exemplos: parques de diversão.*

**Link:** link para o site da sugestão turística, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator.

**Imagem:** link para a imagem principal da sugestão turística, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator.

**Descrição:** breve descrição da sugestão turística, adicionado manualmente pelo administrador, utilizando o aplicativo Aviv Database Populator.

**Indicadores de personalidade:** todos os indicadores gerados pelo Personality Insights (Big Five, Needs, Values e Consumption Preferences)

Usuário:

**Id:** número identificador do usuário, retirado do Facebook. Chave primária.

**Nome:** nome do usuário, retirado do Facebook.

**Gênero:** gênero do usuário, retirado do Facebook.

**E-mail:** e-mail do usuário, retirado do Facebook.

**Indicadores de personalidade:** todos os indicadores gerados pelo Personality Insights (Big Five, Needs, Values e Consumption Preferences)

1. **Comentários sobre a Implementação**

O projeto começou com o objetivo de obter aprendizado e prática em desenvolvimento de aplicativos e ainda não havia conhecimento sobre a linguagem Swift e sobre as APIs do Watson, Facebook e Firebase. Desta forma, foi escolhido um método de desenvolvimento onde os requisitos mudavam constantemente confirme o entendimento dos componentes. Além disso, para chegar no produto final, foram pensados em 5 marcos a serem alcançados, sendo eles:.

1. Login e leitura dos dados do Facebook.
2. Interpretação das postagens do Facebook pelo Personality Insights, gerando um perfil.
3. Adição das sugestões turísticas ao banco de dados.
4. Comparação dos perfis do usuário com a das sugestões turísticas.
5. Apresentação do resultado para o usuário.

Como o aprendizado sobre as tecnologias foi sendo adiquirido de acordo com o desenvolvimento, cada marco teve sua dificuldade particular. Além disso, devido à dependencia entre esses marcos, houve muito retrabalho pelo fato de algum método utilizado em um deles não funcionar bem quando outro foi implementado. A seguir está descrito um pouco da dificuldade de cada um desses marcos.

1. O grande desafio deste primeiro passo foi entender a API do Facebook. A documentação disponivel era difícil de entender e parecia estar desatualizada. Muitos exemplos de iOS ainda estavam em Objective-C ou utilizavam versões antigas da API. Além disso, não havia muita discussão sobre a API em Swift no Google, ou em foruns, Stack Overflow, e outras comunidades.
2. A API do Watson é muito bem explicada e foi fácil entender o Personality Insights. O único problema que aconteceu nesta parte é que o Personality Insights não faz a leitura de textos na língua portugesa, o que me fez ter que escolher um entre dois caminhos: fazer um aplicativo que só compativel com perfis do Facebook de usuários que postam em inglês ou acoplar uma API de tradução linguistica. A primeira opção traz mais desafios de testes a segunda de desenvolvimento. Por fim, decidi escolher utilizar uma API do próprio Watson chamada Language Translator.
3. O ideal era utilizar um webcrawler para ler um website de críticas relacionadas à viagens, enviando as críticas positivas para o Personality Insights. Porém, por questões de complexidade e tempo isto não sería possível neste momento. Então decidi fazer esta mesma ideia de forma manual e utilizando o código que já estava pronto (para a geração do perfil do usuário). Desta forma, criei um aplicativo de apoio, feito para ser utilizado por administradores, para popular o banco de dados. Neste aplicativo, o administrador preenche os campos com dados da sugestão turística e faz login em um perfil de teste do Facebook criado para esta sugestão com postagens retiradas do site de críticas TripAdvisor.
4. Nesta etapa houve um grande desafio por conta do Firebase. Para fazer a comparação dos perfis, é necessário pegar todos indicadores de personalidade, tanto das sugestões turísticas quanto do usuário no banco de dados. O Firestore funciona como uma árvore, onde a raiz é uma *‘collection’* de ‘*documents’*, e cada *‘document’* contem campos com valores ou outra *‘collection’*. O problema é que para fazer a leitura de cada *‘document’*, é necessário abrir uma thread assíncrona e não é possível fazer a leitura de um *‘document’* ao mesmo tempo que outro *‘document’* mesmo que ele esteja dentro de uma *‘collection’* sua. E como a estrutura dos dados estava respeitando o diagrama de classes projetado, ele possuía quase 95 *‘documents’*, ou seja, para o usuário e cada sugestão turísitca, o Firebase abria uma thread assíncrona. Foi perdito muito tempo tentando sincronizar essas threads, até que decidi que o melhor a ser feito sería alterar o que já foi feito, e cheguei à seguinte solução:
   * Não utilizar a base de dados do perfil de personalidade do usuário no Firebase, passando a utilizar apenas os dados locais.
   * Retirar todas as *‘subcollections’* (*‘collections’* dentro de *‘documents’*) das sugestões turísicas e passá-las para o *‘document’* principal.

Esta solução deixou os dados mais desorganizados, diferente do que foi modelado, e custou muito retrabalho. Mas passamos a trabalhar com apenas uma thread assíncrona e isto não apenas facilitou a programação, mas também diminuiu o tempo de resposta dos servires do Firebase e os riscos de bugs.

1. FALTA ESCREVER ESSE AQUI

Um outro desafio da implementação foi que, depois de ficar um tempo sem contato com o projeto, houve uma atualização do Swift 4 para o Swift 4.2. Com isso, os serviços do Watson, que são serviços de computação na nuvem, pararam de dar suporte ao Swift 4. Isto resultou em ter que reescrever completamente o programa, que nesta época estava por volta de 50% pronto. Mas no final foi muito positivo, já que me fez revisar todo o código, fazendo melhorias.

1. **Conclusão**

Falta fazer esta parte

* 1. **Oportunidades para Trabalhos Futuros**
* Webcrawler
* Indicação de cidades (entre as cidades ditas pelo usuário, qual é a mais indicada para ele?)
* Eventos aparecem só quando estão próximos da data de ida
* Algumas indicaçõs (talvez por categoria) podem ter mais peso em alguns indicadores na comparação o perfil do usuário do que outros indicadores.

1. **Referências Bibliográficas**

[1] IBM. **Sugestões de entretenimento usando computação cognitiva.** 2016.

Acesso em: 3/05/2019. Disponível em:

<https://www.ibm.com/developerworks/br/cloud/sugestao-entretenimento-computacao-cognitiva/index.html>

[2] IBM. **Documentação do Personality Insights.** 2019.

Acesso em 25/05/2019. Disponível em:

<https://cloud.ibm.com/docs/services/personality-insights>

**IGNORAR TUDO ABAIXO DESTA LINHA - FAZIA PARTE DO RELATORIO 1**

1. Referências Bibliográficas

[1] – YouTube: IBM Watson: How it Works. Acesso em 20/09/2017. Disponível em:

<https://www.youtube.com/watch?v=_Xcmh1LQB9I&t=3s>

[2] – IBM: Conheça o Watson e seu uso na saúde. Acesso em 20/09/2017. Disponível em:

<https://www.ibm.com/blogs/robertoa/2017/03/conheca-o-watson-e-seu-uso-na-saude/>

[3] – IBM: Beyond bots and robots: Exploring the unrealized potential of cognitive computing in the travel industry. Acesso em 20/09/2017. Disponível em:

<https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=GBE03776USEN&>

[4] – Forbes: Bringwater, Adrian. Come Fly With AI, IBM Cloud Builds ’Chatbot’ Virutal Travel Agent. Acesso em 20/09/2017. Disponível em:

<https://www.forbes.com/sites/adrianbridgwater/2016/11/22/come-fly-with-ai-ibm-cloud-builds-chatbot-virtual-travel-agent/#1088409b4813>

[5] – WayBlazer: Acesso em 20/09/2017. Disponível em:

<https://www.wayblazer.ai/>

[6] – Baseline: Greengard, Samuel. WayBlazer’s Journey Leads to Cognitive Computing. Acesso em 24/09/2017. Disponível em:

<http://www.baselinemag.com/cloud-computing/wayblazers-journey-leads-to-cognitive-computing.html>

[7] – Watson Documentação: Acesso em 20/09/2017. Disponível em:

<https://console.bluemix.net/docs/services/personality-insights/models.html#models>

[8] – Watson Documentação: Acesso em 20/09/2017. Disponível em:

<https://console.bluemix.net/docs/services/conversation/index.html#about>

[9] – Watson Documentação: Acesso em 10/10/2017. Disponível em:

<https://console.bluemix.net/docs/services/conversation/lang-support.html#supported-languages>

[10] – Watson Documentação: Acesso em 10/10/2017. Disponível em:

<https://console.bluemix.net/docs/services/personality-insights/user-overview.html#overviewLanguage>

[11] – Watson GitHub: Acesso em 15/11/2017. Disponível em:

<https://github.com/watson-developer-cloud/swift-sdk/blob/master/docs/quickstart.md>