

INTELIGÊNCIA ARTIFICIAL  
E COMPUTACIONAL

# *Serviços* COGNITIVOS

ERICK GALANI MAZIERO



8

## LISTA DE FIGURAS

Figura 8.1. Exemplo de notebooks para implementação do projeto.....	15
---	----



**LISTA DE TABELAS**

Tabela 8.1. Definição do problema.....	7
Tabela 8.2. Dados disponíveis para o projeto .....	8
Tabela 8.3. Ferramentas indicadas para o desenvolvimento. ....	9
Tabela 8.4. APIs disponíveis para uso no projeto. ....	9
Tabela 8.5. Forma de avaliação .....	11
Tabela 8.6. Disponibilização da solução .....	12
Tabela 8.7. Definição do projeto.....	14

## LISTA DE CÓDIGOS-FONTE

Código-Fonte 8.1. Algoritmo da solução .....	10
--	----

EMSE

**SUMÁRIO**

8 SERVIÇOS COGNITIVOS .....	6
8.1 Definição do projeto.....	6
8.1.1 Definição do problema .....	7
8.1.2 Dados disponíveis .....	8
8.1.3 Ferramentas e APIs .....	8
8.1.4 Desenvolvimento .....	10
8.1.5 Forma de avaliação .....	11
8.1.6 Disponibilização da solução .....	11
8.2 Hands On! .....	14
REFERÊNCIAS.....	16
GLOSSÁRIO .....	17

## 8 SERVIÇOS COGNITIVOS

Agora que você tem conhecimentos da área de Inteligência Artificial e, inclusive, já desenvolveu algumas aplicações pequenas e ilustrativas usando técnicas de Aprendizado de Máquina, que tal desenvolver um projeto na área? Vamos fazer um passo a passo para você atacar um problema, desde a sua definição até o desenvolvimento de uma solução, sua avaliação e disponibilização.

É muito importante a definição inicial do escopo de um projeto, pois atacar (sair desenvolvendo) um problema sem algumas definições iniciais é como atirar sem um alvo, querendo atingir muita coisa, sem sucesso. Além disso, por se tratar de um projeto em que visaremos o uso de técnicas de Aprendizado de Máquina, a disponibilidade de dados é essencial, principalmente em uma abordagem supervisionada. Mesmo em uma abordagem não supervisionada, os dados são necessários, ainda que não anotados de alguma forma. Por fim, como você avaliará o sucesso de sua solução? Definir uma forma de avaliação é importante, seja ela intrínseca ou extrínseca.

A seguir, são apresentadas diversas etapas e em algumas delas podem aparecer conceitos e técnicas com as quais talvez você não esteja familiarizado. Portanto, gaste um tempo pesquisando a respeito e tente implementar!

### 8.1 Definição do projeto

Ao mesmo tempo que você será guiado na definição do seu projeto, um exemplo será apresentado para auxiliá-lo a entender cada passo. Depois dessa seção, você poderá iniciar o desenvolvimento da solução com maiores chances de sucesso. Vale salientar que, no momento do desenvolvimento, alguns ajustes na definição serão necessários, aumentando ou diminuindo o escopo, modificando os dados, ferramentas, etc.

### 8.1.1 Definição do problema

Bom, vamos partir da definição do problema. Como exemplo, considere a tabela “Definição do problema”.

Tabela 8.1. Definição do problema

Problema	Diversos aplicativos para telefones celulares são disponibilizados com problemas até então desconhecidos por seus desenvolvedores.
Lacunas	Os testes unitários e funcionais elaborados pelos desenvolvedores podem não cobrir todo o código criado ou todas as funcionalidades do aplicativo. Portanto, outras formas de identificar problemas devem ser elaboradas.
Hipóteses	A análise dos comentários e revisões dos usuários em uma loja virtual de aplicativos, o Google Play, pode indicar problemas não identificados pelos desenvolvedores.
Objetivos	Analisar os comentários e revisões de aplicativos com técnicas de PLN (Processamento da Linguagem Natural), mais especificamente análise de sentimentos, com o intuito de levantar possíveis problemas, para auxiliar os desenvolvedores na melhoria da qualidade do software desenvolvido.

Fonte: Desenvolvido pelo autor (2018)

A tabela “Definição do problema” apresenta a definição de um problema clássico em desenvolvimento de softwares. Nem sempre os desenvolvedores conseguem fazer o teste completo de todo o código e funcionalidades de um aplicativo para smartphones, por exemplo. Essa é uma lacuna que pode ser melhor elaborada com técnicas de Inteligência Artificial em larga escala. Então, define-se uma hipótese em que se acredita que a análise de comentários a respeito de um aplicativo em uma loja virtual possa conter relatos de problemas encontrados pelos usuários e que não sejam de conhecimento dos desenvolvedores.

Baseado nessa hipótese, traça-se como objetivo específico o uso de uma técnica de PLN chamada Análise de Sentimentos, para identificar possíveis problemas

em aplicativos, por meio da análise dos comentários em uma loja virtual, o Google Play.

### 8.1.2 Dados disponíveis

Para atingir o objetivo definido, temos que verificar se temos dados disponíveis, assim como certificar que tais dados têm licença que possibilita o uso. No caso dos comentários do Google Play (Tabela 8.2), a Google disponibiliza o acesso de forma gratuita por meio de uma API (<https://cloud.google.com/apis/>).

Tabela 8.2. Dados disponíveis para o projeto

<b>Dados disponíveis</b>	Comentários ( <i>Reviews</i> ) dos aplicativos na Google Play.
<b>Licença</b>	Acesso livre e gratuito

Fonte: Desenvolvido pelo autor (2018)

Cada comentário será classificado em positivo ou negativo pela análise de sentimentos. Os comentários negativos serão selecionados para identificar as palavras-chave (substantivos). Esses substantivos podem indicar o problema encontrado pelo usuário, ou a funcionalidade que não tem o comportamento esperado.

Uma versão mais rebuscada desse projeto poderia ser com o uso da análise de sentimentos voltada a aspectos, que em um mesmo comentário identifica partes do texto que são positivas e partes negativas. Para cada parte, identifica a entidade avaliada, assim como o aspecto da entidade. Por exemplo, a interface de um aplicativo pode ser uma entidade e o tamanho da fonte, um aspecto, que pode ser positivo ou negativo.

### 8.1.3 Ferramentas e APIs

Como ferramentas (Tabela 8.3), vamos definir a linguagem de programação, seus pacotes e APIs que serão utilizadas para atingir o objetivo do projeto. Como de costume, vamos utilizar a linguagem Python, interpretada no Jupyter. Para disponibilizar a solução na forma de uma API Rest, utilizaremos o Flask.



Tabela 8.3. Ferramentas indicadas para o desenvolvimento

<b>Ferramentas</b>	Python, Jupyter, Flask
--------------------	------------------------

Fonte: Desenvolvido pelo autor (2018)

Como APIs (Tabela 8.4) utilizaremos a API do Google para acesso aos dados. Para realizar a análise de sentimentos, utilizaremos a API do Watson para o processamento da linguagem natural (NLU – *Natural Language Understanding*), que, dentre as várias análises, faz a análise de sentimentos.

Para cada API é importante saber a forma de acesso, se exige algum tipo de autenticação. No caso das duas APIs citadas, ambas exigem acesso com autenticação, uma com conta Google (<https://developers.google.com/android-publisher/authorization>), outra, IBM.

Outra informação que se deve saber é a precificação. No caso da API da Google, para acesso aos comentários, é necessário um registro como desenvolvedor no Google Play com o pagamento de uma tarifa (<https://play.google.com/apps/publish/signup/#ApiAccessPlace>). Já no caso da IBM Watson, são possíveis apenas 30 mil itens NLU por mês no plano gratuito. Para entender um item NLU, segundo o console da API (<https://cloud.ibm.com/catalog/services/natural-language-understanding>), um NLU consiste em um texto de 10 mil caracteres e uma extração, por exemplo, a identificação de entidades.

Tabela 8.4. APIs disponíveis para uso no projeto

<b>APIs</b>	Google, IBM Watson
<b>Precificação</b>	<i>Google</i> : exige registro de desenvolvedor <i>IBM Watson</i> : 30.000 itens NLU por mês
<b>Acesso</b>	<i>Google</i> : com autenticação <i>IBM Watson</i> : com autenticação

Fonte: Desenvolvido pelo autor (2018)

#### 8.1.4 Desenvolvimento

Agora que já sei o que vou utilizar, tenho que definir meus algoritmos. Como busco os dados, em que momento chamo a API. Como lido com os retornos e como gero as minhas respostas.

No exemplo de análise de sentimentos (AS) de comentários, o seguinte algoritmo de alto nível poderia ser definido no Código-Fonte 8.1.

```
Obtenha o package_name (id do app) da requisição
Busque os comentários sobre o app
Para cada comentário faça
    Pegue o texto do comentário e chame a API de AS
    Se for positivo, guarde na lista_positivos
    Senão, guarde na lista_negativos
Para cada comentário em lista_positivos faça
    Pegue os substantivos e armazene em subst_positivos
    Faça a contagem da frequência dos substantivos
    Ordene a contagem decrescente
Para cada comentário em lista_negativos faça
    Pegue os substantivos e armazene em subst_negativos
    Faça a contagem da frequência dos substantivos
    Ordene a contagem decrescente
Gere a resposta do serviço
Retorne a resposta da requisição
```

Código-Fonte 8.1. Algoritmo da solução  
Fonte: Desenvolvido pelo autor (2018)

O uso de Inteligência Artificial está encapsulado na chamada à API do Watson, que faz todo o trabalho de PLN. Caso essa solução não atenda aos requisitos, principalmente após a avaliação da solução (próxima subseção), o ideal é fazer o treinamento de um novo modelo de PLN que faça a AS com bons resultados para o problema.

Cada um dos passos do Código-Fonte 8.1 contém diversos outros passos, omitidos aqui por se tratar de um exemplo para você desenvolver a sua solução, para o seu problema.

A solução proposta será disponibilizada por meio de um serviço REST, conforme será explicado a seguir.

### 8.1.5 Forma de avaliação

Para fazer a avaliação da solução, serão selecionados, manualmente, 20 comentários obtidos manualmente de um mesmo aplicativo, sendo 10 positivos e 10 negativos. Para definir um comentário positivo ou negativo, serão utilizadas as estrelas (*score*) dos comentários, validados manualmente. A validação manual é necessária, pois um comentário pode ter 5 estrelas, mas ser negativo, e vice-versa.

O ideal é que os comentários positivos e negativos sejam validados por mais de uma pessoa, de forma que haja concordância entre os validadores. Por questão de simplicidade, você mesmo pode fazer isso.

Tabela 8.5. Forma de avaliação

<b>Dados de teste</b>	Criação de um conjunto de teste com 20 comentários. Esse conjunto deve ser balanceado, com 10 comentários positivos e 10 negativos. Seleção dos substantivos dos comentários.
<b>Avaliação intrínseca</b>	Executar a solução sobre os dados de teste e apresentar as métricas de acurácia, precisão e cobertura.

Fonte: Desenvolvido pelo autor (2018)

Na Tabela 8.5 é apresentada uma avaliação intrínseca da solução projetada. Uma avaliação intrínseca considera a solução como um fim, ou seja, por meio de algumas métricas, verifica o quão correto está seu comportamento. No exemplo, quantos dos exemplos positivos e negativos são corretamente identificados com a solução proposta. Caso tenha alguma dúvida com relação às métricas de avaliação, retorne ao capítulo de aprendizagem de máquina e veja algumas métricas que foram apresentadas.

### 8.1.6 Disponibilização da solução

Feito o desenvolvimento da solução, como tornar essa implementação facilmente acessível aos usuários? O exemplo apresentado neste capítulo pode ser útil para os desenvolvedores de aplicativos, que podem consultar as análises dos comentários de seus aplicativos em busca de melhorias.

Uma forma de tornar simples e fácil o acesso a uma solução é disponibilizá-la na forma de um serviço REST (*REpresentational State Transfer*). Essa forma de

acessar aplicações tem sido amplamente utilizada e possibilita a interoperabilidade de sistemas diversos.

Tabela 8.6. Disponibilização da solução

<b>Disponibilização</b>	A solução será disponibilizada através de um serviço REST
<b>Endpoints</b>	<code>http://meuservico:8080/ api/analyzereviews/package_name</code>

Fonte: Desenvolvido pelo autor (2018)

Para a linguagem utilizada em nossa solução exemplo, o Python, temos um *framework* chamado Flask (<https://palletsprojects.com/p/flask/>) que permite a criação de serviços REST a partir de seu código Python. Essa é a forma de disponibilização do serviço indicada no projeto exemplo, conforme a Tabela 8.6Disponibilização da solução”.

Quando se define um serviço REST, faz-se necessário disponibilizar cada funcionalidade da solução por meio de um *endpoint*. No exemplo, temos apenas um *endpoint*, definido como `http://meuservico:8080/api/analyzereviews/package_name`, visto que definimos uma funcionalidade em que simplesmente é feita a análise de comentários de um dado aplicativo.

Quando o serviço é executado, ele fica disponível para acesso na rede e geralmente é necessário definir uma porta (8080). O final do *endpoint* (`package_name`) deve ser substituído pelo código do aplicativo que se deseja analisar, por exemplo, `http://meuservico:8080/api/analyzereviews/com.spiffygame`.

Para acessar esse *endpoint*, o usuário deve fazer uma requisição HTTP em qualquer linguagem de programação, ou através de um browser qualquer. O retorno da requisição, geralmente, é feito no formato JSON (*Javascript Object Notation*) que é equivalente ao dicionário, do Python. Você deve, então, definir a estrutura do retorno, com suas chaves e valores. Por exemplo:

```
{ "analyzereviews":  
  
  {  
  
    "package_name": "com.spiffygame",
```

```
    "number_reviews": 10,  
    "positive_reviews":  
    {  
        "number": 3,  
        "most_common_nouns": ["jogabilidade", "gráficos", ...]  
        "reviews": [  
            "Texto do comentário 1...",  
            "Texto do comentário 2 ...",  
            ...  
        ]  
    }  
    "negative_reviews":  
    {  
        "number": 7,  
        "most_common_nouns": ["botões", "travamento", ...]  
        "reviews": [  
            "Texto do comentário 1...",  
            "Texto do comentário 2 ...",  
            ...  
        ]  
    }  
}  
}
```

## 8.2 Hands On!

Agora que já foram apresentados e ilustrados os passos para a definição do projeto, é sua vez de projetar! Comece preenchendo a tabela a seguir (Tabela 8.7). Lembre-se de que, durante a implementação, algumas informações poderão mudar.

Tabela 8.7. Definição do projeto

<b>Problema</b>	
Problema	
Lacunas	
Hipóteses	
Objetivos	
<b>Dados</b>	
Dados disponíveis	
Licença	
<b>Ferramentas e APIs</b>	
Ferramentas	
APIs	
Precificação	
Acesso	
<b>Avaliação</b>	
Dados de teste	
Avaliação intrínseca	
<b>Disponibilização</b>	
Disponibilização	
Endpoints	

Fonte: Desenvolvido pelo autor (2018)

Agora, utilizando a linguagem Python no Jupyter, você já pode começar a criar seus notebooks, conforme ilustrado na imagem a seguir. Lembre-se que o Jupyter notebook dá a facilidade de verificar a execução de cada trecho de código logo após o código, numa mesma tela, inclusive com a utilização de gráficos.

Já a disponibilização de sua solução será feita em um ou mais arquivos python (.py) que será chamado pelo Flask para a criação de um serviço REST, tornando seus *endpoints* disponíveis na rede.

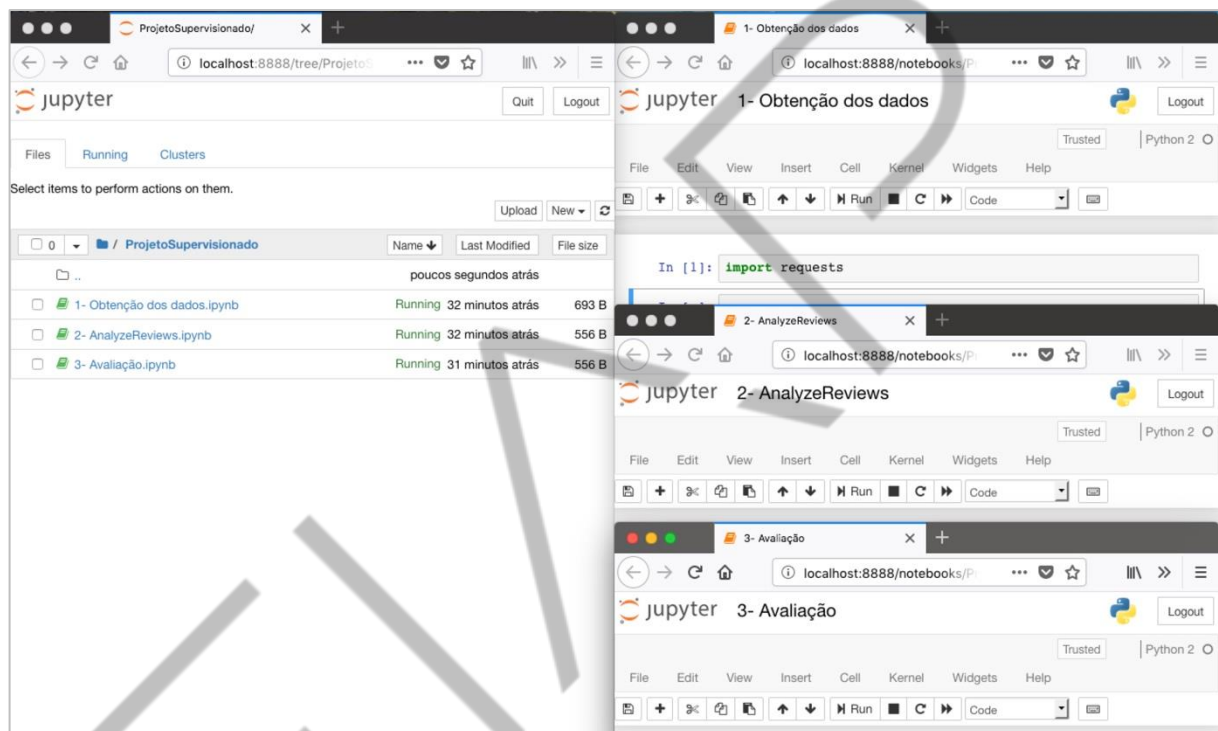


Figura 8.1. Exemplo de notebooks para implementação do projeto  
Fonte: Desenvolvido pelo autor (2018)

Há muito a falar sobre cada um dos passos abordados neste capítulo, mas com o que foi apresentado você já consegue ter uma boa noção de como desenvolver um projeto com Inteligência Artificial e pode vislumbrar o caminho a seguir para criar suas próprias soluções com IA.

## REFERÊNCIAS

GOOGLE PLAY DEVELOPER API. **Reviews**. [s.d.]. Disponível em: <<https://developers.google.com/android-publisher/api-ref/reviews>>. Acesso em: 26 jul. 2018.

IBM. **Api explorer**. [s.d.]. Disponível em: <<https://developer.ibm.com/api/list>>. Acesso em: 26 jul. 2018.

EMANIP



## GLOSSÁRIO

<b>Endpoint</b>	Endereço (URL) de um serviço disponível para acesso por uma aplicação.
-----------------	--

EMANIP