UNIVERSIDADE FEDERAL DE PERNAMBUCO CENTRO DE INFORMÁTICA GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO - CIN UFPE

Uma software que ajuda empresas de TI a encontrar os melhores candidatos, cruzando datasets do GitHub e StackOverflow

Aluno: Matheus de Souza Lins {msl@cin.ufpe.br}
Orientador: Vinicius Cardoso Garcia {vcg@cin.ufpe.br}
Avaliador: Leandro Maciel Almeida {lma3@cin.ufpe.br}
Área: Data Science e Engenharia de Software

Recife, Dezembro de 2018

Resumo

O crescimento de informação e a quantidade de dados na internet é evidente. E para que as empresas de TI, principalmente as quais possuem dificuldades em encontrar profissionais qualificados[1] - segundo pesquisa realizada pelo IBGE[3] - , não fiquem para trás, é preciso filtrar toda essa informação.

Recrutadores técnicos de grandes empresas procuram bons profissionais. E para que essa procura não se torne uma missão árdua e cansativa, este trabalho mostra a construção de uma solução que pode ajudar empresas a escolherem o candidato mais adequado à sua vaga.

Através de um algoritmo simples de classificação, foi possível destacar percentualmente os melhores perfis de uma das plataformas de código mais usadas no mundo, o GitHub, com oportunidades de emprego em outra plataforma onde a comunidade de TI é extremamente presente, o StackOverFlow.

Por outro lado, a solução não só beneficiará o recrutador, mas também os usuários do GitHub. Através de gráficos, os *GitHub users* terão um resumo de quais tecnologias estão sendo mais usadas no momento e também uma visão percentual de quando o seu perfil se encaixa em determinada oportunidade de emprego.

Palavras-chave: Empresas de TI, Recrutador Técnico*, Data Science,* GitHub, StackOverflow, Emprego e Tendências Tecnológicas.

Abstract

Growing of information and how many data exists in the internet are unstoppable. And, for what IT Companies, mainly ones that has difficulties to find goods professionals, doesn't stand back, is necessary to filter all these informations.

Tech recruiters and big companies search qualified professionals and to this searching do not become a hard and tiring search, this job shows a building of a platform that can help organizations choose the most suitable candidate for their position.

Through an simple classification algorithmic it was possible to classify by percentage the bests profiles from one of them code platforms more used of the world, the GitHub, with jobs opportunities in another one where IT community is extremely present, the StackOverFlow.

In other side, the solution will not only benefit the recruiter, but GitHub users too. Through graphics, they will have a report of which ones technologies are being more used in the moment and a percentual vision about how mutch their GitHub profile match with a specific job opportunity.

Keys-Word: IT Company, Tech Recruiter*, Data Science,* GitHub, StackOverflow, Job and Technology Trends.

Sumário

1. **Introdução**
   - 1.1. Contexto
   - 1.2. Motivação
   - 1.3. Solução
   - 1.4. Método
2. **Obtenção dos Dados**
   - 2.1. GitHub
   - 2.2. StackOverflow
3. **Limpeza e Armazenamento dos dados**
4. **Exploração dos dados**
   - 4.1. Métricas
5. **Modelo dos dados**
   - 5.1. O sistema de classificação
   - 5.2. O Algoritmo básico
   - 5.3. Ponderamento
   - 5.4. Algoritmo final
6. **A Plataforma (Solução)**
7. **Resultados**
8. **Trabalhos Futuros**
9. **Referências**
10. **Glossário**

1. Introdução

1.1 Contexto

O crescimento de dados na internet é contínuo. Segundo um estudo realizado pelo portal Statista, o número de usuários nas redes sociais no ano de 2021 pode atingir três bilhões de usuários[16] e a expectativa é que seja gerado mais de quatrocentos bilhões de gigas de dados diariamente no ano de 2025[17].

Com tanta informação, é muito difícil encontrar em poucos segundos o que se procura. Existem milhares de links e fontes. Em suma, não conseguimos discernir (em pouco tempo) qual a melhor loja de roupas ou o melhor lugar para se visitar nas férias. É preciso procurar com cautela para não cair em nenhum *clickbait*[1] ou fraude. Por mais que tenhamos a maior ferramenta de pesquisa online do mundo (o Google), ainda é difícil ter velocidade na resposta.

Os dados que crescem todos os dias, são classificados em estruturados ou não estruturados[4] e podem ser gerados por qualquer dispositivo *smart* com acesso à internet. Denomina-se *BigData* [5] o conjunto de dados muito grande ou complexo.

Um número crescente de empresas vêm empregando a filosofia de *BigData Analytics*[6] para ajudá-las a gerar *insights* e tomar sempre a melhor decisão. Adquirir respostas valiosas acerca das tendências de mercado e principalmente conhecer mais sobre o comportamento de seus clientes, seus anseios e expectativas, são cenários os quais grande massa de dados - que se transformam em informação - ajudam a validar.

Em alguns setores a concorrência é altíssima e os negócios que não se adaptarem a essa mudança radical do mercado, estarão sujeitos a desaparecer em pouco tempo; e é óbvio que nenhuma empresa quer ficar para trás. Por isso, a busca constante por profissionais especializados está cada dia mais aguçada.

As consequências da má escolha

Nenhuma empresa quer contratar o funcionário errado, principalmente quando o cargo é crucial para os objetivos estratégicos. Ao fazer uma má escolha, não só a empresa perde tempo, gasto de recursos e esforços, mas também o funcionário (que será demitido) pode manchar sua imagem diante do mercado. No final das contas, ambas as partes saem perdendo.

A empresa por sua vez é a que mais se prejudica, pois, na maioria das vezes, tenta manter o funcionário por mais tempo até perceber que realmente ele não se encaixa no perfil do cargo e isso pode lhe custar caro.

O índice de insatisfação no Brasil é alto. Nove em cada dez gestores [4], acreditam que já contrataram funcionários que não atenderam às suas expectativas.

1. É um termo pejorativo que se refere à conteúdo da internet destinado à geração de receita de publicidade on-line, normalmente às custas da qualidade e da precisão da informação. Em: https://pt.wikipedia.org/wiki/Clickbait

1.2 Motivação

Principalmente para pequenas empresas que estão começando no mercado de TI, possuem bastante dificuldade em encontrar o profissional compatível com o seu momento crítico de incerteza.

As chamadas *startups* - organizações que têm em média 10 a 20 funcionários - em sua maioria não possuem um modelo de negócio bem definido e por isso não conseguem estabilidade financeira no mercado. Consequentemente o capital para se manter é curto, o que leva aos gestores convencerem os bons profissionais da área a acreditarem no futuro do negócio mesmo recebendo pouco.

A motivação que encontrei para realizar esse trabalho se resume na ajuda dessas empresas iniciantes para que possam saber mais sobre o candidato em um dos pilares mais importantes da contratação, **o técnico**.

1.3 Solução

Com o foco neste problema, esse trabalho propõe criar uma ferramenta que facilitará a pesquisa que recrutadores técnicos têm quando procuram o melhor funcionário para sua companhia de TI.

A partir de cruzamentos de dados extraídos de *datasets* do GitHub, o usuário da plataforma, terá uma visão da probabilidade em percentagem de quando um perfil é compatível com a vaga de emprego que está sendo oferecida no StackOverFlow. Assim, as chances de se fazer uma má escolha diminuem assim como o tempo de procura.

O principal objetivo da ferramenta **é mostrar o quanto a vaga se encaixa com as características do candidato de acordo com seu histórico de uso do GitHub**. Ou seja, a empresa terá mais uma comprovação de que as habilidades que o candidato descreve na candidatura são habilidades que ele realmente domina. Vale salientar que esta solução é apenas mais uma forma de ajudar no recrutamento, não substitui nem automatiza nenhum processo seletivo.

A aplicação foi pensada de uma forma que o recrutador técnico poderá definir quais filtros ele deseja comparar entre as duas bases. Ou seja, o sistema não será pré-determinado, pelo contrário, será manuseado de uma forma altamente dinâmica.

De um lado, o GitHub nos dar uma gama de variáveis que podem ser acessadas de um determinado perfil: projetos, *commits*, *pull requests*, *stars*, *forks*, *followers* e etc. De outro lado, o StackOveflow possui informações sobre oportunidades de emprego (*jobs)*[5] em todo o mundo. Cada *job* possui variáveis que podem ser comparadas com as do GitHub, como: tecnologias da vaga, nível de experiência exigido, cidade, benefícios e etc.

Se juntarmos esses dados, do GitHub de um determinado usuário e uma oportunidade de emprego do StackOverflow, teremos um "*match* perfeito" que agrada os dois lados, a empresa e o candidato.

A empresa poderá entrar na plataforma com o seu perfil *Business* do StackOverflow, imediatamente o sistema reconhece quais suas vagas cadastradas e atualiza a base. Em seguida, é possível escolher para qual vaga se deseja *rankear* os candidatos de uma determinada região. Pronto, o sistema colocará, em

ordem de relevância, os melhores candidatos para a vaga. Caso o usuário queira refinar a busca, ele poderá trocar, adicionar ou remover os filtros que são gerados automaticamente ao escolher a vaga.

Do outro lado, o candidato também poderá acessar a plataforma através do seu perfil do Github e verá as oportunidades que mais combinam com ele, aplicando determinados filtros.

1.5 - Método

Para que fosse encontrada a melhor forma de fazer o *"match perfeito"* entre a vaga e o candidato, seguiu-se um passo a passo simples de *Data Science* chamado OSEMN, um acrônimo em inglês que significa:

- **O** - *Obtaining our data* (Obter os dados)
- **S** - *Scrubbing / Cleaning our data* (Limpar os dados)
- **E** - *Exploring* (Explorar os dados)
- **M** - *Modeling* (Modelar os dados)
- **N** - *Interpreting* (Interpretar os dados)

Esta *pipeline* não é um segmento oficial mas um que me chamou bastante atenção em um artigo que li[2] e que me norteou. Não segui a risca o que o autor mostra, mas uma grande parte foi aproveitada.

Baseado nessa ideia, este documento está dividido da seguinte forma:

1. **Obtenção dos dados**: Onde mostro a riqueza dos dados que serviram como base para a elaboração do experimento e como consegui obtê-los.
2. **Limpeza e armazenamento dos dados (Pré-Processamento)**: Mostro como utilizei ferramentas para enxugar tanta informação retirada na extração.
3. **Exploração dos dados**: Após a limpeza, é importante entender o que temos em mãos. Nesta etapa foi feito algumas análises dos dados baseadas em métricas.
4. **Modelo dos dados**: A parte onde a "mágica" acontece. Foi criado um

algoritmo que determina qual *job* é ideal para o perfil do github.

5. **Resultados:** Onde será mostrado exemplos de perfis após a aplicação do algoritmo e quais *insights* pode-se tirar.

2. Obtenção dos dados

O Github é a maior plataforma de compartilhamento de código do mundo. São mais de 24 milhões de usuários, 1,5 milhões de organizações, 67 milhões de repositórios e mais de 1 bilhão de commits desde Setembro de 2016 [5].

Do outro lado, o StackOverflow é a maior plataforma *Q&A (Question and Answer)* na área de TI do mundo. Em um dia comum são cadastradas mais de 8 mil perguntas no site sobre problemas de programação [6].

Não é surpresa para nós que essas duas plataformas estão revolucionando a forma de se relacionar com código e a agilidade na programação. Mas o que as duas têm em comum? É possível obter seus dados. Ambas as empresas possuem *REST API's* [7]. O GitHub por sua vez, ainda disponibiliza os dados de outras formas, através do *Google BigQuery*, do site *GHTorrent , GitHub Archive e GitHub Contents* [8]*.* Já o Stackoverflow, além de ser um fórum, possui um espaço só para oportunidades de emprego (Jobs, como eles chamam), e é justamente com esses *Jobs* que irei trabalhar.

Para fins de capacidade de armazenamento e processamento, foi feito um filtro dos jobs com as principais cidades no ramo de tecnologia do mundo incluindo algumas do Brasil. No total foram escolhidas 30 cidades, são elas:

1	Amsterdam, Holanda	11	Curitiba, Brasil	21	Nova York, Estados Unidos
2	Bangalore, India	12	Dublin, Irlanda	22	Quebec, Canadá
3	Belo Horizonte, Brasil	13	Estocolmo, Suécia	23	Recife, Brasil
4	Barcelona, Espanha	14	Florianópolis, Brasil	24	Rio de Janeiro, Brasil
5	Berlim, Alemanha	15	Fortaleza, Brasil	25	São Francisco, Estados Unidos

6	Boston, Estados Unidos	16	Hong Kong, China	26	São Paulo, Brasil
7	Bruxelas, Bélgica	17	Londres, Inglaterra	27	Seoul, Coréia do Sul
8	Budapeste, Hungria	18	Moscow, Russia	28	Tóquio, Japão
9	Campinas, Brasil	19	Manaus, Brasil	29	Toronto, Canadá
10	Copenhague, Dinamarca	20	Natal, Brasil	30	Vancouver, Canadá

Tabela 1: Cidades tecnológicas escolhidas para o experimento

2.1 GitHub

Os dados aqui já são bastante estruturados por se tratarem de dados extraídos de uma API. Por isso, para obter as informações de um usuário, foi preciso seguir 3 passos:

1. Criar um OAuth app na sessão para Desenvolvedores
2. Salvar duas chaves de segurança (secret_key e secret_id).
3. Fazer uma requisição para um endpoint passando as chaves.

O endpoint que foi utilizado para a captura das informações do usuário, retorna seus principais dados junto a outros endpoints, como pode-se ver abaixo:


```
{
    "login": "matheuslins",
    "url": "https://api.github.com/users/matheuslins",
    "html_url": "https://github.com/matheuslins",
    "followers_url": "https://api.github.com/users/matheuslins/followers",
    "following_url": "https://api.github.com/users/matheuslins/following{/other_user}",
    "gists_url": "https://api.github.com/users/matheuslins/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/matheuslins/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/matheuslins/subscriptions",
    "organizations_url": "https://api.github.com/users/matheuslins/orgs",
    "repos_url": "https://api.github.com/users/matheuslins/repos",
    "events_url": "https://api.github.com/users/matheuslins/events{/privacy}",
    "received_events_url": "https://api.github.com/users/matheuslins/received_events",
    "type": "User",
    "site_admin": false,
    "name": "Matheus Lins",
    "company": null,
    "blog": "linkedin.com/in/matheus-lins-a9592b99/",
    "location": "Blumenau - SC - Basil",
    "email": null,
```

```
    "hireable": null,
    "bio": "I'm a Software Developer who loves to work with data. Python Programmer and
Evangelist. Coffee transformer in code that makes technology evolve",
    "public_repos": 25,
    "public_gists": 4,
    "followers": 15,
    "following": 22,
    "created_at": "2014-06-19T03:40:15Z",
    "updated_at": "2018-11-30T01:18:40Z"
}
```


Imagem 1: Exemplo da resposta da API de um usuário

2.2 StackOverflow

A princípio não foi possível acessar a API da plataforma para capturar os *Jobs*, porém esses dados foram capturados.

Para montar o *dataset* do stackoverflow, criou-se um *robô crawler* que captura as principais informações dos *jobs*. Para a criação do mesmo, foi utilizado o framework Scrapy, um projeto *open source* em Python que facilita e muito a criação de *Web Crawlers.* Com isso, foram capturados um total de **6715** itens.

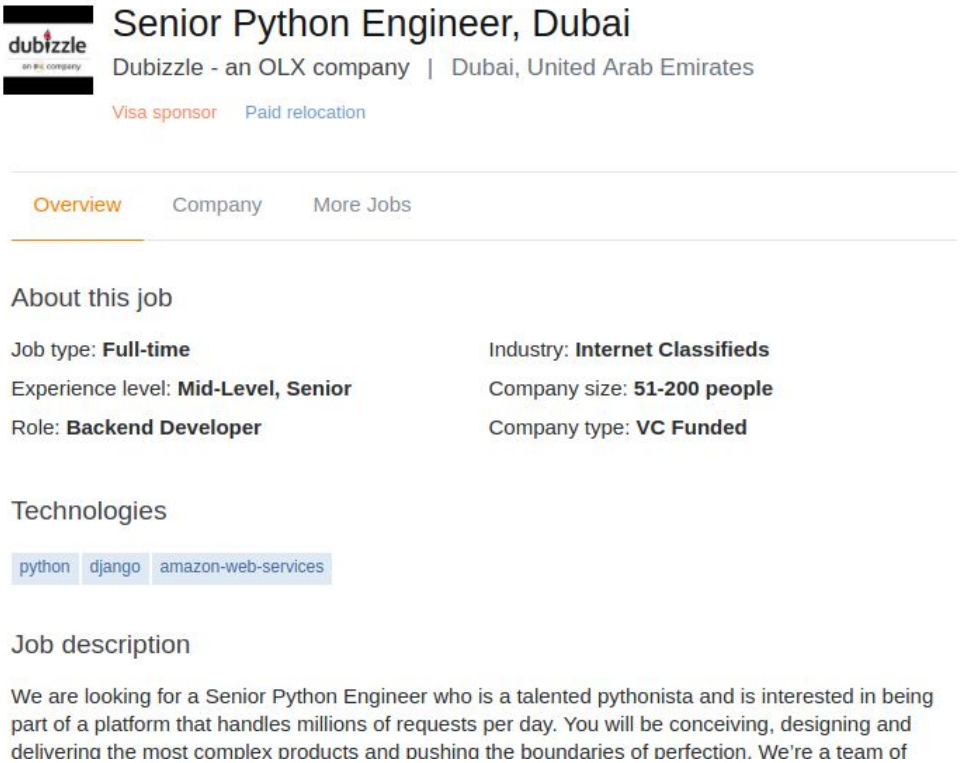

Imagem 2: Exemplo de Job no StackOveflow

```
import re
import os

from furl import furl
from decouple import config
from scrapy.spiders import Spider

from .constants.consulta import START_URL
from .steps.consulta import consult_job


class StackOverflowSpider(Spider):
    name = 'stkflow'
    city = None
    distance = None
    job = None
    allowed_domains = ['stackoverflow.com']
    custom_settings = {
        'DOWNLOAD_DELAY': 0.8
    }

    def __init__(self, city=None, distance=None, job=None, *a, **kw):
        self.city = city
        self.distance = distance
        self.job = job
```

Imagem 3: Código do Robô Crawler, mais em: <https://github.com/linkehub/crawlpy/tree/master/crawlpy/crawlpy/spiders/stackoverflow>

3. Limpeza e Armazenamento dos dados

Após os dados coletados, foi preciso fazer alguns pré-processamentos (apenas nas oportunidades de emprego do StackOverflow, já que os dados do GitHub já são bem estruturados.

Imagem 4: Exemplo de um Job no site do StackOverflow que foi salvo na base.

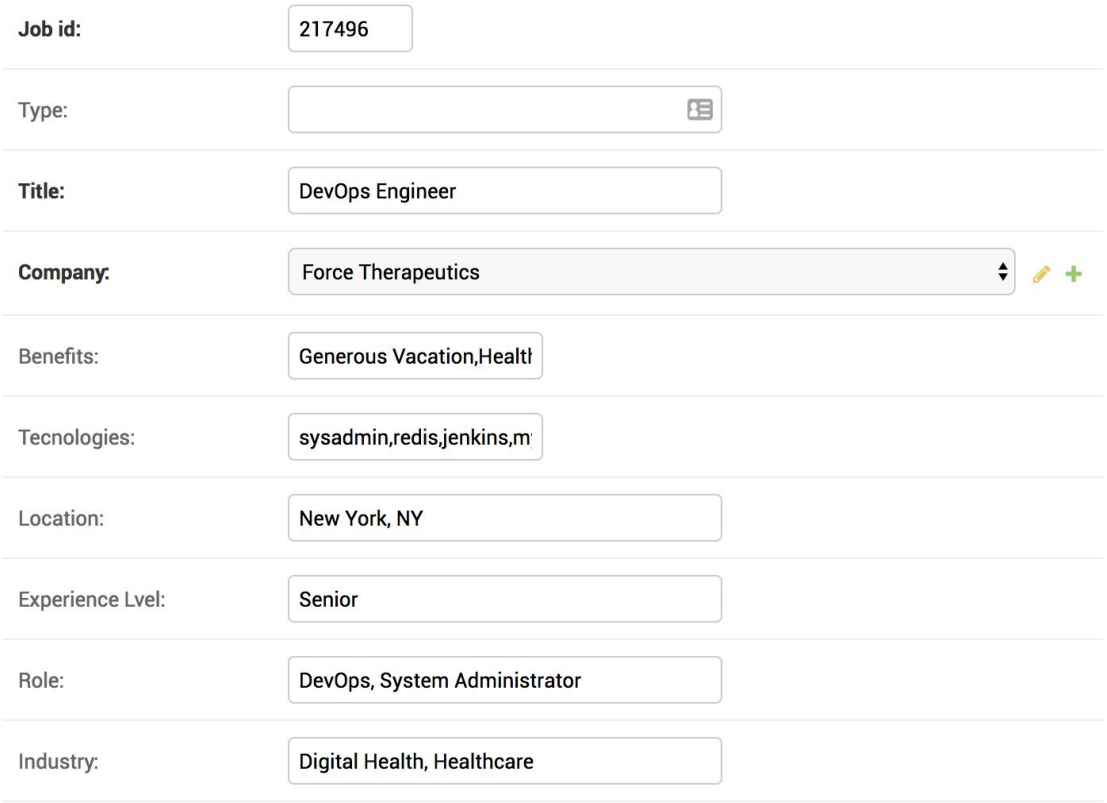

Imagem 5: Como o job foi salvo na base

Logo em seguida, os mesmos foram armazenados no PostgresSQL, um banco robusto e rápido tanto na leitura, quanto na escrita. Como as informações dentro da solução final precisam ser bem estruturadas, um banco relacional foi escolhido.

Analisando quais campos dos dois *datasets* seriam necessários para a exploração dos dados, chegou-se ao seguinte estado:

Job	Perfil do GitHub
Título	Login
Empresa	Localização
Benefícios	Linguagens mais fortes a cada repositório
Tecnologias Exigidas	Nomes dos repositórios
Localização	*Forks* dos repositórios
Nível de experiência exigido	*Watchers* dos repositórios
Cargo	*Stars* dos repositórios
Indústria	Se o repositórios é *forkado*
Salário	Quantidade de issues reportadas em cada repositório
Descrição	Quantidade de contribuidores em cada repositório
Url	

Tabela 2: Campos escolhidos para o experimento

4. Exploração dos dados

Uma vez que as informações sobre o perfil do Github do usuário são extraídas, determina-se algumas métricas que ajudam na construção do algoritmo.

4.1 Métricas

Para cada perfil, são identificadas as principais tecnologias utilizadas nos repositórios e com base nesta lista, são extraídas as seguintes métricas:

Do lado do GitHub:

1. Quantidade de **arquivos da tecnologia** escritos nos repositórios
2. O **número de repositórios** criados com uma determinada tecnologia.
3. Quantidade de **estrelas** de todos os repositórios com a tecnologia.
4. Quantidade de ***forks*** de todos os repositórios com a tecnologia.
5. Quantidade de ***watchers*** de todos os repositórios com a tecnologia.
6. Quantidade de ***issues reported*** de todos os repositórios com a tecnologia.
7. Quantidade de ***contributors*** de todos os repositórios com a tecnologia.

Do lado do StackOverflow:

1. **Linguagens de programação exigidas** da vaga
2. O **nível de experiência exigido**

5. Modelo dos dados

Com as métricas extraídas de cada dataset, chegou-se a um modelo padrão que determina a porcentagem de combinação entre uma determinada vaga de emprego e um perfil de um usuário do GitHub.

Tal modelo baseia-se em um sistema de pontuação ponderado, o que determina o nível de sucesso de um perfil em relação a vaga. Com isso, se torna viável determinar numericamente a combinação.

5.1 - O Sistema de Classificação

Assim como *score* no setor crédito, cada métrica impulsiona a pontuação do perfil de um usuário do GitHub, o que leva a encaixá-lo em uma determinada categoria. As métricas do perfil são comparadas junto ao seu respectivo peso e o

perfil ganha uma classificação por tecnologia. Basicamente, o usuário pode se classificar em quatro categorias, são elas:

1. **Iniciante**: De 0 a 200 pontos
2. **Júnior**: De 201 a 400 pontos
3. **Pleno**: De 401 a 600 pontos
4. **Sénior**: De 601 a 800 pontos
5. **Especialista**: De 801 a 1000 pontos

5.2 - O Algoritmo Básico

Foi pensado de uma forma simples, mas que de fato entregasse valor ao usuário da plataforma.

Para que seu funcionamento fique claro, abaixo listo o passo a passo:

1. Ler todos os repositórios públicos do usuário
2. Determinar quais tecnologias tem em cada repositório
3. Mapear uma determinada tecnologia para quantidade de ocorrência em cada repositório, além da quantidade de linhas escritas da mesma
4. Criar um ranking das tecnologias mais usadas pelo usuário
5. Após isso, pergunta-se: quais são as vagas de emprego que possuem as mesmas tecnologias requeridas que o usuário tem em seu ranking?
6. Criar um mapeamento da tecnologia para a lista de empregos que combinam.

Porém, apenas estes 6 passos não são o suficiente para determinar o melhor candidato para uma vaga de emprego. Pensando nisso, foi adicionado mais um passo, que de fato usa do ponderamento das métricas para a classificação e *rankeamento* das melhores vagas.

5.3 - Ponderamento

Cada métrica possui um peso, e estes pesos foram definidos arbitrariamente por conversão. Não existe algo que determine o que torna um perfil do github mais bem sucedido do que outro. Então, após pesquisar em fóruns da comunidade, encontrei um bem interessante[15] que justificou os pesos que foram dados para cada métrica. Em ordem decrescente, os pesos são os seguintes:

1. **Contribuidores (22)**:Quando se trata de um repositório *open source* que possui muitas pessoas contribuintes, é claro que algo chama a atenção, é valioso para a comunidade.

2. **Issues Reportadas (18):** Quando alguém reporta algum problema no repositório, significa que ela se deu o trabalho de entender o que ali estava escrito, o que consequentemente torna-o mais relevante.

3. ***Forks* (16):** É um indicador que mostra que alguém se interessou bastante pelo projeto ou já está usando e precisa fazer modificações. Ou seja, ela está resolvendo seu problema usando o determinado repositório ou no mínimo usará futuramente.

4. ***Stars* (14):** Mostra que no mínimo pessoas gostaram do propósito do repositório. Além de que possa ser que já estejam usando.

5. **Watchers (13):** Determina o quanto se tem o interesse no determinado projeto, já que, quem é watcher recebe notificações de qualquer alteração no mesmo.

6. **Repositórios da linguagem (10)**: Aparentemente quanto mais repositórios o usuário tem em uma determinada linguagem, mas ele tem conhecimento sobre ela. Porém, não necessariamente isso é certo, já que podem existir repositórios clonados em seu perfil ou códigos em uma determinada

linguagem que não foram feitos por ele. Mas mesmo assim, esta é uma métrica que merece atenção.

7. **Arquivos escritos (7):** A que tem um peso menor, mas que é importante ser considerada, pois é preciso analisar se de fato o repositório tem arquivos escritos na tecnologia determinada.

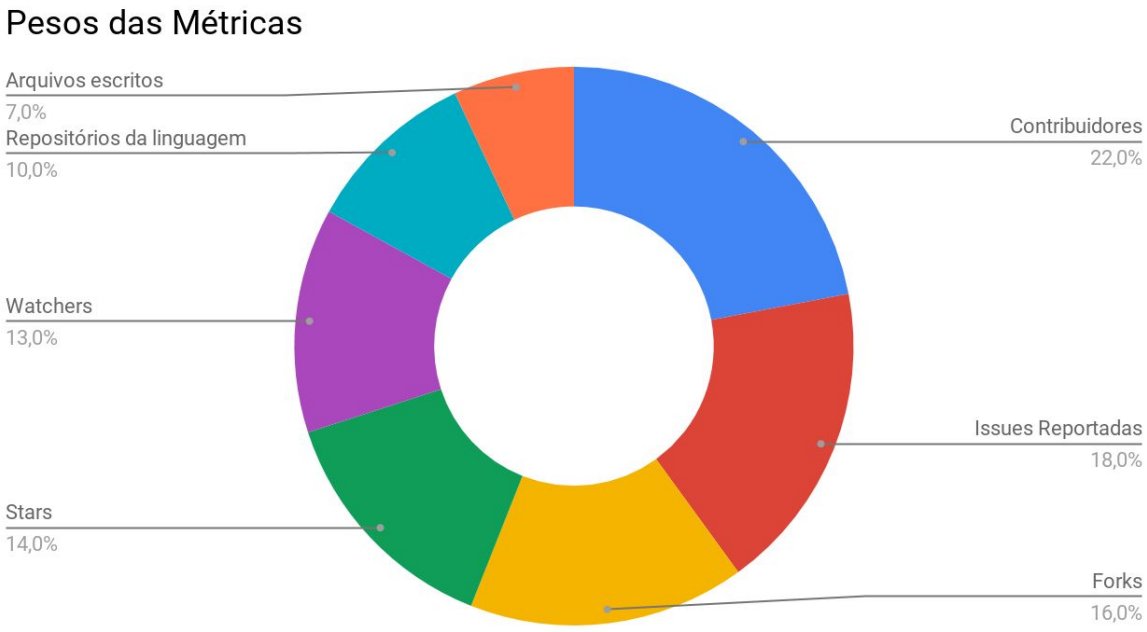

Gráfico 1: Pesos das Métricas

Com a definição dos pesos e a contagem das métricas do perfil, é feita a média ponderada, o que irá determinar a pontuação em cada linguagem. Consequentemente a pontuação definirá o nível do usuário, que finalmente se classificará em um usuário Iniciante, Júnior, Pleno, Sênior ou Especialista. Abaixo, é mostrado um exemplo:

Suponhamos que um determinado perfil tenha os seguintes números na linguagem **Python**:

Stars	Forks	Repositórios	Watchers	Issues	Contribuintes	Arquivos
5	3	10	2	1	5	150

Se aplicarmos o ponderamento nesses números, chegamos ao seguinte resultado:

Stars	5	x	14	=	70	/ 7	203
Forks	3	x	16	=	48	Júnior	
Repositórios	10	x	10	=	100		
Watchers	2	x	13	=	26		
Issues	1	x	18	=	18		
Contribuintes	5	x	22	=	110		
Arquivos	150	x	7	=	1050		

Chegamos a conclusão que este usuário é do nível Júnior na linguagem Python. Com esta simples fórmula, é possível classificar todo o usuário do github dentro da plataforma.

5.4 - Algoritmo final

Após o último passo do algoritmo básico e a análise ponderada das métricas do perfil, foi preciso aplicar mais um passo. Ao invés de apenas mapear as oportunidades de emprego de acordo com as linguagens comparando as quais o perfil do usuário tinha, mapeou-se também comparando com mais uma métrica da vaga, o nível exigido.

Pois, após o processo de ponderamento, se tornou possível saber o nível do perfil.

6. A plataforma (Solução)

A solução foi pensada para atender dois públicos alvo, qualquer usuário que tenha uma conta no GitHub, e um recrutador técnico. Um usuário comum do GitHub possui números personalizados de acordo com os dados de seu perfil (como dito no tópico 5 - Modelo de dados), já um técnico recrutador possui informações (*insights*) que poderá vir ajudá-lo na melhor escolha.

Dados do usuário do GitHub:

- As tecnologias/linguagens de programação mais quentes de acordo com as métricas definidas na explicação do tópico anterior

Suas Top Linguagens

- CSS - 209, 10
- HTML - 160, 10
- Python - 158, 11
- JavaScript - 34, 7
- Dockerfile - 15, 1
- Shell - 8, 1
- Makefile - 0, 6
- PHP - 0, 1

- A pontuação a cada tecnologia/linguagem de programação

Seu score a cada linguagem

- CSS - 231
- HTML - 182
- Python - 196
- JavaScript - 51
- Dockerfile - 24
- Shell - 17
- Makefile - 31
- PHP - 1

- O Nível de programação do usuário por tecnologia/linguagem de programação

Seu nível a cada linguagem

- CSS - Júnior
- HTML - Iniciante
- Python - Iniciante
- JavaScript - Iniciante
- Dockerfile - Iniciante
- Shell - Iniciante
- Makefile - Iniciante
- PHP - Iniciante

- As vagas de emprego que mais combinam com o seu perfil

Jobs para voce

CSS

- Lead Front-End Developer that wants to save Earth! - Boulder, CO
- Front End UI Developer - Morrisville, NC
- STAFF UI SOFTWARE ENGINEER - PHP and Javascript - San Jose, CA
- Senior Frontend Engineer - Zürich, Switzerland
- Student/part-time Software Developer - Wien, Austria

HTML

- Fullstack Developer - JavaScript / Java - Berlin, Deutschland
- Front End UI Developer - Morrisville, NC
- STAFF UI SOFTWARE ENGINEER - PHP and Javascript - San Jose, CA
- Front-end Developer with Angular expertise - New York, NY
- Web-Entwickler (m/w/d) PHP/MySQL - Tarp, Deutschland
- Student/part-time Software Developer - Wien, Austria

Python

- Data Backend Engineer (Python) - Retail Operations - Berlin, Deutschland
- Cybersecurity and Technology Control Product TCO - Columbus, OH
- Graduate Systems Engineer - Chicago, IL
- Cybersecurity and Technology Control Product TCO - Tampa, FL
- Agile Coach - Software Engineering - Columbus, OH
- Senior Data Scientist - Stockholm, Sweden

Dados que o recrutador virá:

- A lista de usuários de acordo com os seguintes filtros
    - Cidade
    - Nível de experiência em uma determinada tecnologia/linguagem de programação
- Quais usuários mais combinam com suas vagas cadastradas

7. Resultados

Para a construção da plataforma a apuração de resultados claros, usei o meu perfil do github (github.com/matheuslins) como porta de entrada.

Levando em consideração que que apenas o lado do usuário do GitHub está pronto, chegou-se os seguintes resultados:

1. **Linguagens mais usadas/populares:**

O primeiro número representa a quantidade de arquivos que existem em todos os repositórios de acordo com a linguagem. Já o segundo, mostra em quantos repositórios aquela linguagem aparece.

- CSS - 209, 10
- HTML - 160, 10
- Python - 158, 11
- JavaScript - 34, 7
- Dockerfile - 15, 1
- Shell - 8, 1
- Makefile - 0, 6
- PHP - 0, 1

2. **Pontuação a cada linguagem**

A pontuação é o resultado de todo o cálculo ponderado mostrado na seção de modelagem de dados.

- CSS - 231
- HTML - 182
- Python - 196
- JavaScript - 51
- Dockerfile - 24
- Shell - 17
- Makefile - 31
- PHP - 1

3. **O nível do usuário a cada linguagem**
    - CSS - Júnior
    - HTML - Iniciante
    - Python - Iniciante
    - JavaScript - Iniciante
    - Dockerfile - Iniciante
    - Shell - Iniciante
    - Makefile - Iniciante
    - PHP - Iniciante

4. ***Jobs* que combinaram com o perfil do usuário**

Separados por tecnologia, além de mostrar o local.

- **CSS**
    - Lead Front-End Developer that wants to save Earth! - Boulder, CO
    - Front End UI Developer - Morrisville, NC
    - STAFF UI SOFTWARE ENGINEER - PHP and Javascript - San Jose, CA

    - Senior Frontend Engineer - Zürich, Switzerland
    - Student/part-time Software Developer - Wien, Austria
    - Senior Frontend Javascript Engineer - Zalando Lounge - Berlin, Germany
- **HTML**
    - Fullstack Developer - JavaScript / Java - Berlin, Deutschland
    - Front End UI Developer - Morrisville, NC
    - STAFF UI SOFTWARE ENGINEER - PHP and Javascript - San Jose, CA
    - Front-end Developer with Angular expertise - New York, NY
    - Web-Entwickler (m/w/d) PHP/MySQL - Tarp, Deutschland
    - Student/part-time Software Developer - Wien, Austria
    - Senior Frontend Javascript Engineer - Zalando Lounge - Berlin, Germany
    - (Senior) Frontend Developer (m/w) - Köln, Deutschland
- **Python**
    - Data Backend Engineer (Python) - Retail Operations - Berlin, Deutschland
    - Cybersecurity and Technology Control Product TCO - Columbus, OH
    - Graduate Systems Engineer - Chicago, IL
    - Cybersecurity and Technology Control Product TCO - Tampa, FL
    - Agile Coach - Software Engineering - Columbus, OH
    - Senior Data Scientist - Stockholm, Sweden

Após a aplicação real do a algoritmo, foi perceptível que o mesmo ainda possui algumas falhas além de não conseguir aprender ao longo do tempo. No meu caso, a linguagem que mais sei é Python e o nível que estou é Pleno. O resultado final me mostra que meu nível é Iniciante, o que, ao meu ver, não retrata tanto a realidade.

Por outro lado, como já dito, foi criada uma solução para que sirva como mais uma fonte de dados no processo seletivo de um programador. Ou seja, podemos concluir que as expectativas foram atingidas, de fato, os números podem ajudar um recrutador técnico em sua escolha e ainda pode servir - em algumas situações - como diferencial na hora da contratação.

8. Trabalhos Futuros

Este trabalho possui um objetivo de ajudar a selecionar os melhores perfis para uma determinada vaga de emprego na área de TI, mais especificamente uma vaga de desenvolvedor de software.

No momento em que esta solução foi apresentada para a banca, a mesma[21] possuía apenas o que foi descrito, porém, é evidente que ainda existe uma abertura para a implantação de muitos outros recursos.

O primeiro ponto a se pensar na evolução da plataforma, é fazer com que o algoritmo aprenda ao longo do tempo. A cada recrutador ou usuário do GitHub, aplicando-se Machine Learning, o sistema se tornaria mais inteligente.

Técnicas como rede neural ou regressão linear, se encaixariam muito bem neste tipo de problema que é prever uma nova informação baseado em dados prévios.

Assim, ficam alguns pontos que na minha visão são os mais importantes para a evolução da solução:

1. Aplicação de uma rede neural para uma melhor dedução dos números finais.
2. Aplicação de um algoritmo de linguagem natural na descrição dos jobs, para que se tenha um maior entendimento sobre ele e consequentemente deixe a análise (ao comparar com o perfil do GitHub) mais eficiente.
3. Criação de gráficos para facilitar o entendimento dos números
4. Criação da área do recrutador técnico na plataforma.

9. Referências

[1] **Profissionais de TI: os desafios na contratação**: Disponível em: <http://www.administradores.com.br/artigos/carreira/profissionais-de-ti-os-desafios-na-contratacao/72795/>

[2] **A Beginner's Guide to the Data Science Pipeline:** Disponível em**:** <https://towardsdatascience.com/a-beginners-guide-to-the-data-science-pipeline-a4904b2d8ad3>

[3] **Pesquisa do IBGE (Pag 26)**: Disponível em: <https://ww2.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000011300312192012031212741739.pdf>

[4] **BigData, o que é e por que é importante?** Disponível em: <https://www.sas.com/pt_br/insights/big-data/what-is-big-data.html>

[5] **BigData por Wikpédia:** Disponível em: <https://pt.wikipedia.org/wiki/Big_data>

[6] **Big Data Analytics: você sabe o que é?** Disponível em: <http://www.bigdatabusiness.com.br/voce-sabe-o-que-e-big-data-analytics/>

[7] **Como uma contratação errada afeta a empresa.** Disponível em: <https://epocanegocios.globo.com/Inspiracao/Carreira/noticia/2015/05/como-uma-contratacao-errada-afeta-empresa.html>

[8] **Stackoverflow Jobs**. Disponível em: <https://stackoverflow.com/jobs>

[9] **Dados estatísticos do GitHub.** Disponível em: <https://octoverse.github.com/>

[10] **Dados estatísticos do StackOveflow.** Disponível em: <https://stackoverflow.blog/2017/05/09/introducing-stack-overflow-trends/>

[11] **What is API REst**. Disponível em: <https://www.quora.com/What-is-a-REST-API>

[12] **Recursos do GitHub**. Disponível em: <https://github.com/fhoffa/analyzing_github>

[13] **The 13 Best Tech Cities in the World (2018).** Disponível em: <https://www.careeraddict.com/best-tech-cities>

[14] **The 25 most high-tech cities in the world.** Disponível em: <http://www.businessinsider.com/the-most-high-tech-cities-in-the-world-2016-6>

[15] **What metric on GitHub is the most telling of a successful open-source project?.** Disponível em: <https://www.quora.com/What-metric-on-GitHub-is-the-most-telling-of-a-successful-open-source-project>

[16] **Number of social network users worldwide from 2010 to 2021 (in billions).** Disponível em: <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>

[17] **How Much Data is Created on the Internet Each Day?.** Disponível em: <https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/>

[18] **10 most difficult IT jobs for employers to fill.** Disponível em: <https://www.cio.com/article/3279767/hiring-and-staffing/10-most-difficult-it-jobs-for-employers-to-fill.html>

[19] **Infographic: The real cost of hiring the wrong Employee**. Disponível em: <http://www.bodhiresourcing.com/real-cost-hiring-wrong-employee/>

[20] **Datasets, o que são e como utilizá-los.** Disponível em: <https://www.aquare.la/datasets-o-que-sao-e-como-utiliza-los/>

[21] **Repositório com o projeto**. Disponível em: <https://github.com/matheuslins/cuscuzjobs>

10. Glossário

1. **Datasets**: Um conjunto de dados que possui uma determinada característica temporal. Exemplo, várias linhas em uma tabela em uma banco de dados que registram informações de quando um determinado usuário entrou no site e o que ele fez dentro do mesmo.
2. **Robô Crawler:** Um algoritmo capaz de capturar os dados de uma determinada página web.
3. **Open Source:** Um software ou ferramenta que possui seu código aberto.

Assinaturas

Matheus de Souza Lins
Orientando

Vinicius Cardoso Garcia
Orientador