

Recomendação Content-Based na Globo.com

Rodrigo Kassick

linkedin.com/in/rkassick

globo.com



THE
DEVELOPER'S
CONFERENCE

Guia

- Recomendação na Globo.com
- Recomendação Baseada em Conteúdo
- Content-based para Usuário
- Próximas Ações



4K

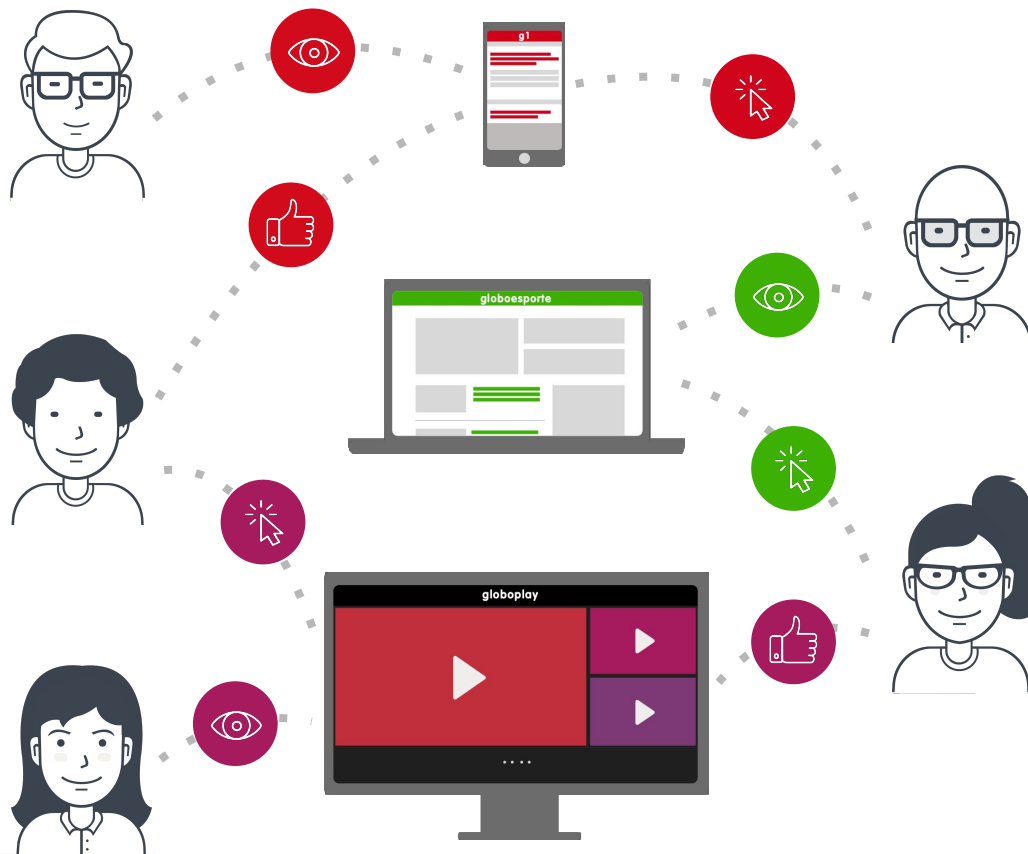
conteúdos novos por dia

30 Mi

usuários únicos
por dia

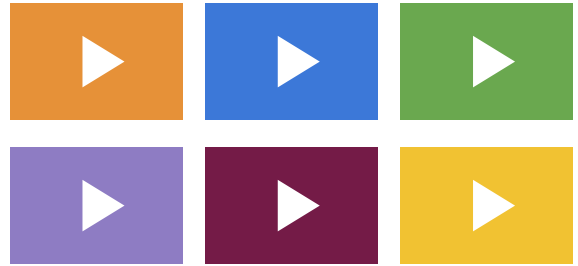
5 Bi

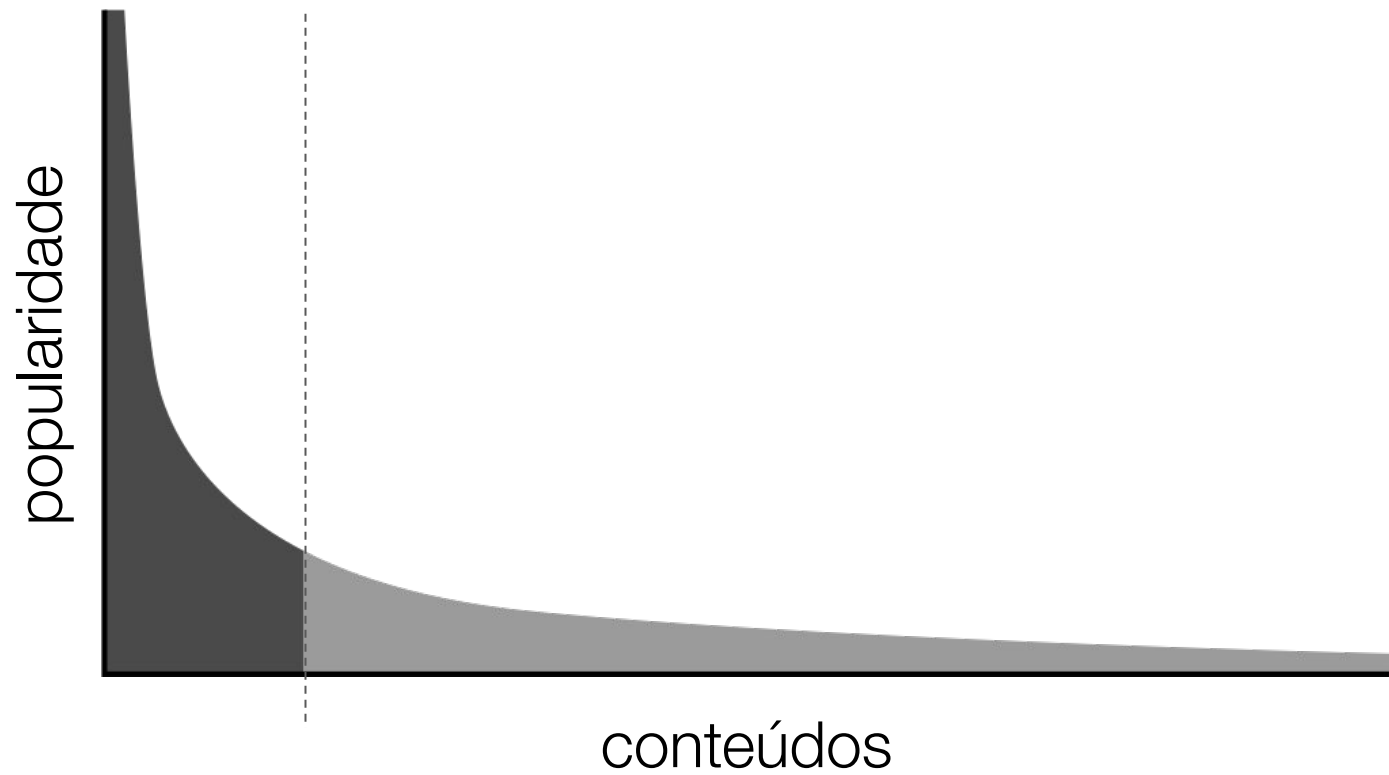
interações usuário-conteúdo
por dia









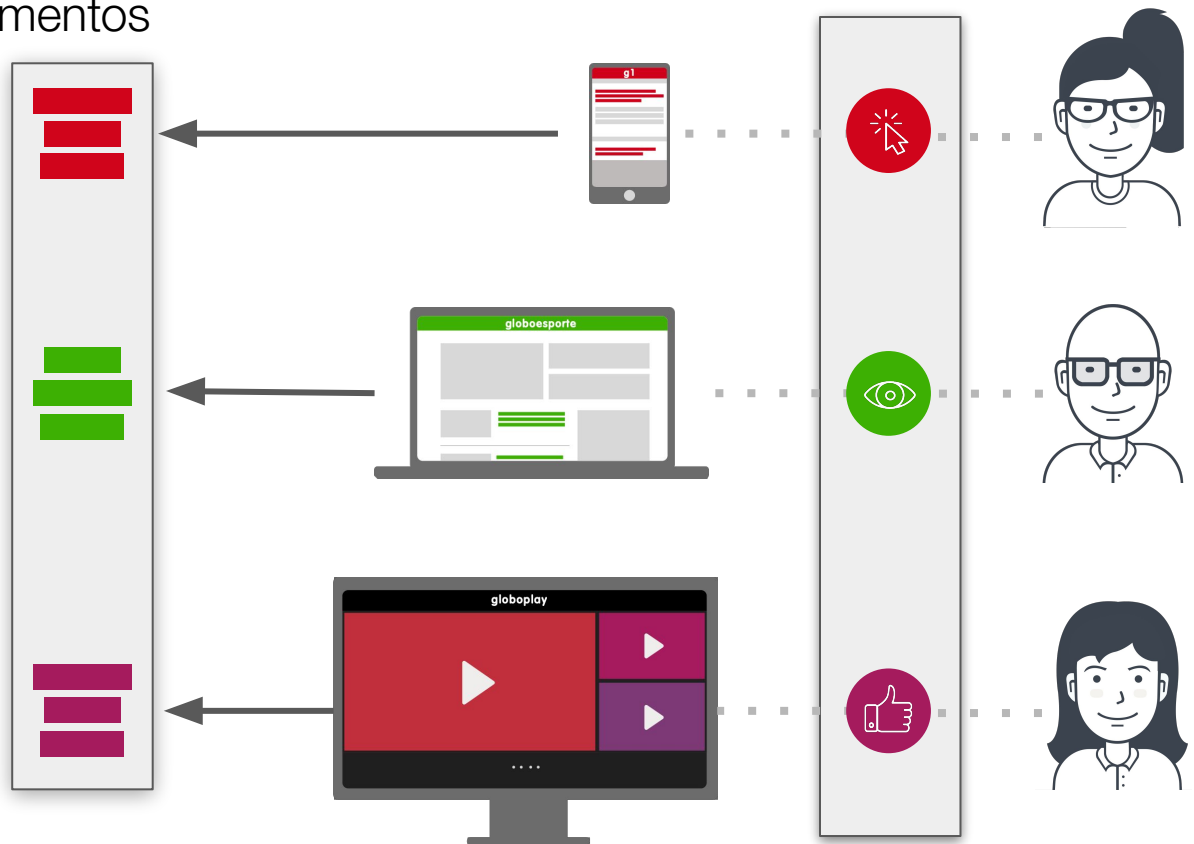






Recomendação

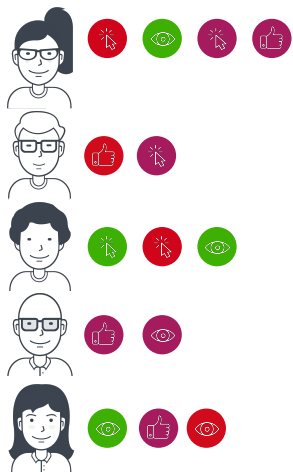
meta-dados de documentos



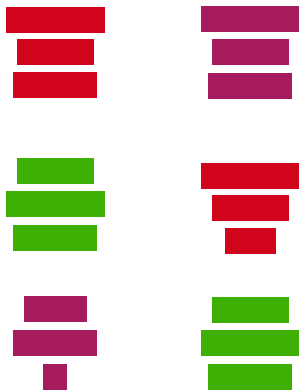
sinais de consumo

Recomendação

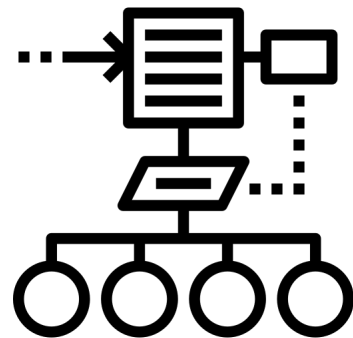
Histórico de Uso



Dados de Documentos



Algoritmos



Famílias de Algoritmos



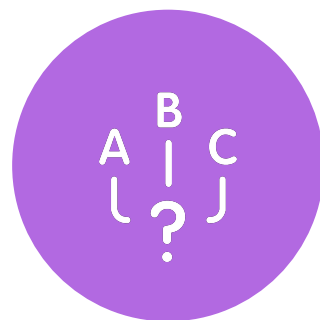
Top



Baseado em
Conteúdo

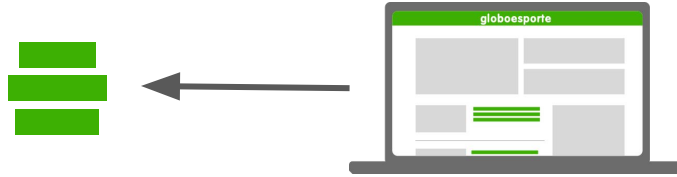


Filtragem
Colaborativa



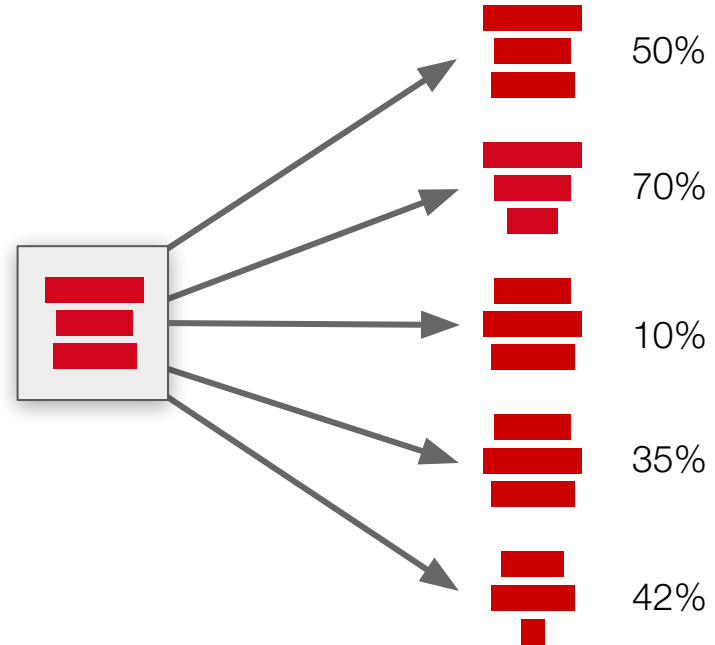
Bandits

Recomendação Baseada em Conteúdo

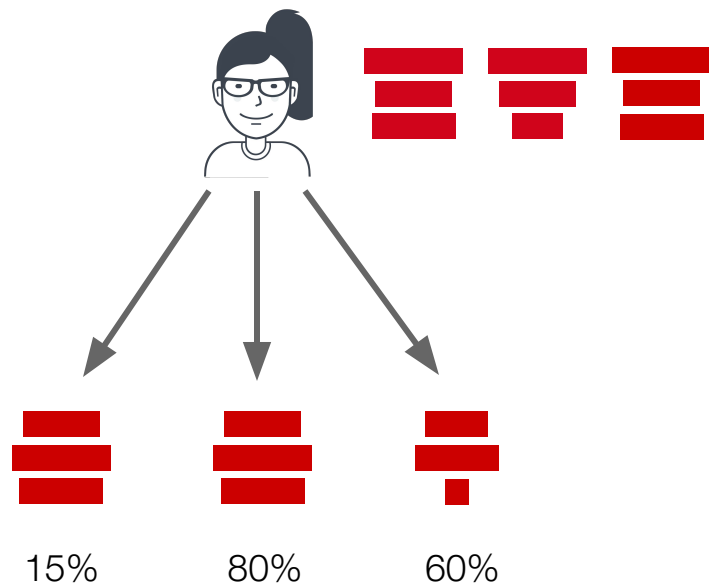




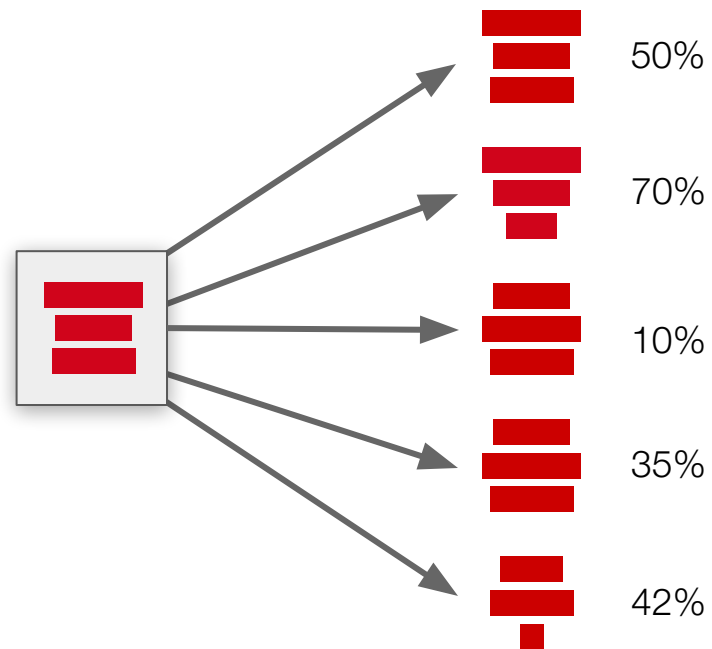
Abordagem para Item

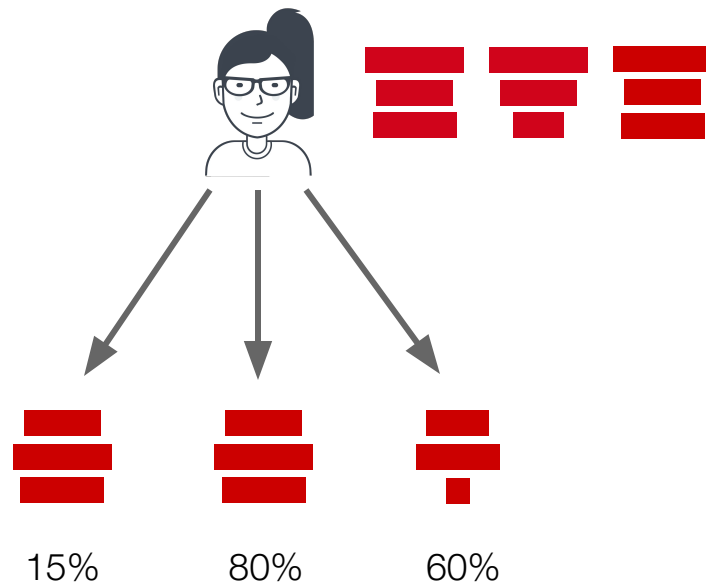


Abordagem para Usuário



Abordagem para Item





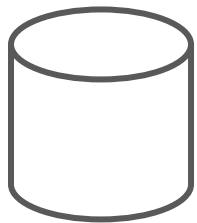
Content-Based para Usuário

Content-Based para Usuário

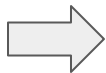
- Perfil de Usuário
 - Gerar Representação do Usuário a partir dos *documentos consumidos*
- Recomendação
 - Encontrar documentos *similares ao perfil de usuário*
 - Gerar *Recomendação* a partir dos documentos similares
- API de Recomendação
 - Entrega resultados previamente computados

Fonte de Meta-dados: Solr

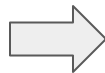
- Documentos armazenados em base **Solr**
- Texto **tokenizado**, sem *stop-words*
- Score de **relevância** de cada **token**
 - **tf·idf**: Relevância de um *termo*
 - **tf**: Frequência de um termo em um **documento**
 - **idf**: Frequência Inversa do termo no **corpus**



Pageview
 $h-1$



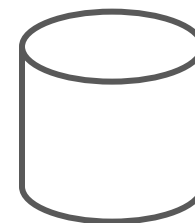
APACHE
Spark



Usuário



Termo	Score
<i>fgts</i>	0.9
<i>previdência</i>	0.5
<i>senado</i>	0.6



Perfis
de Usuários

doc



term + score

Solr

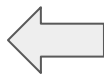
user



perfil



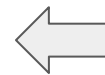
Documentos
Recomendados



Consulta por
Termos

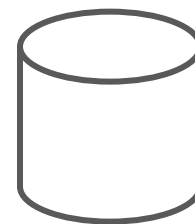
term + score

docs +
relevância



Recomendação
para Usuário

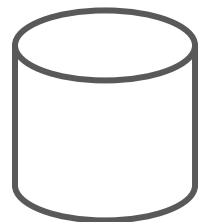
Termo	Score
<i>fgts</i>	0.9
<i>previdência</i>	0.5
<i>senado</i>	0.6



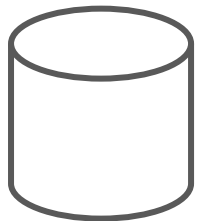
Perfis
de Usuários

Primeiro Objetivo: **Tempo de Resposta da API**

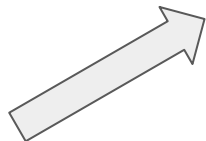
- Resultados para usuário **já computados**
- Fonte de Meta-dados: Lucene + Solr



Pageview
d-1



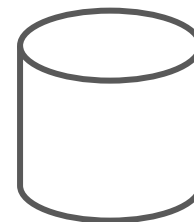
Perfis
de Usuários
d-1



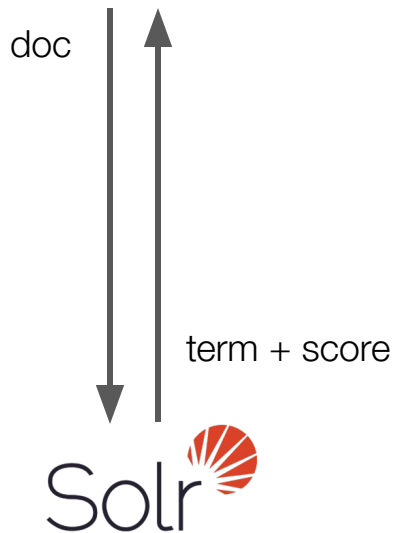
Usuário

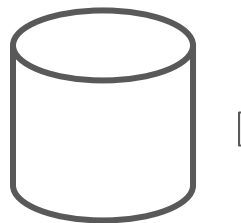


Termo	Score
<i>fgts</i>	0.9
<i>previdência</i>	0.5
<i>senado</i>	0.6

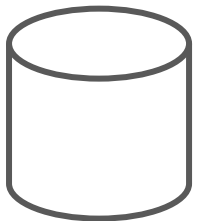
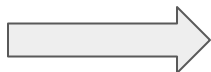


Perfis
de Usuários
d-0

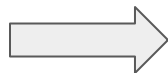
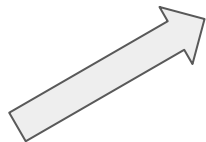




Documentos



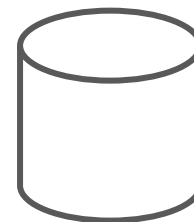
Perfis
de Usuários
d-0



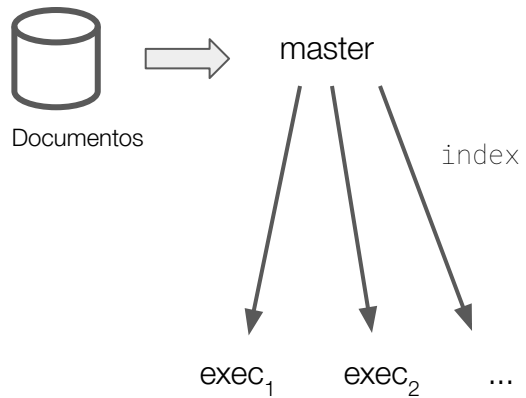
Usuário



Documento	Score
<i>doc1</i>	0.9
<i>doc2</i>	0.7
<i>doc3</i>	0.65



Recomendações
Computadas



```
index = lucene.index(docs)

broadcast(index)

userProfiles.map { user =>
  index.similar(user.terms)
}
```

Resultados

- CTR similar a Filtragem Colaborativa para Usuário
- Geração de novos perfis de usuário em ~4h
- Geração de novas recomendações em ~1h

Content-Based: **Meta-dados Melhores**

Content-Based: **Meta-dados Melhores**

- Solr/Lucene:

“Chuva no Rio de Janeiro causa alagamentos”

Termo	Score
chuva	0.5
rio	0.6
janeiro	0.7
causa	0.3
alagamento	0.6

Segundo Objetivo: **Meta-dados Melhores**

- Solr/Lucene: Tokenização não ideal
- **Aurelio**
 - Solução de Representação de **Conteúdo Textual**
 - Baseada em **Redes Neurais**
 - Identificação de **Termos Compostos** (*campos do jordão, presidente da república, etc.*)
 - Representação do documento como *vetor de termos* com score **tf·idf**

Usuário



Termo	Score
trânsito	0,65
chuva	0,4

	chuva	trânsito
doc1	0,7	0,1
doc2	0,3	0,5
doc3	0,3	0,6

chuva

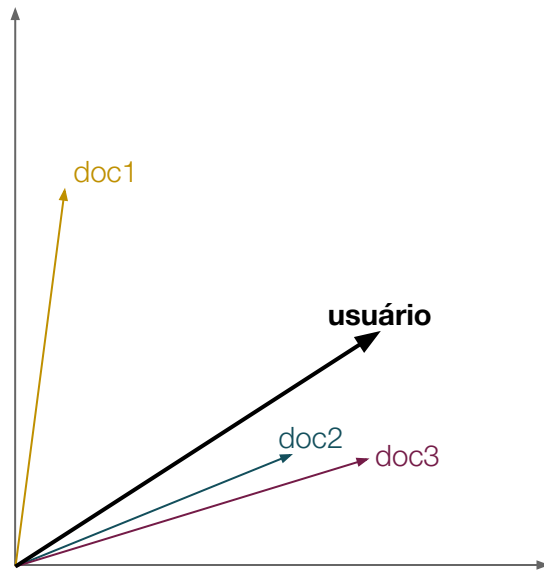
doc1

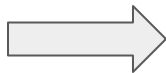
usuário

doc2

doc3

trânsito

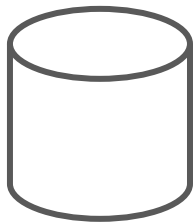




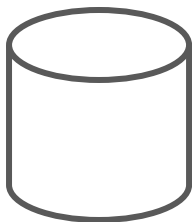
Usuário



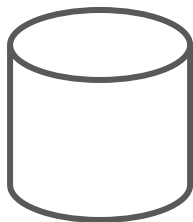
Termo	Score
<i>fgts</i>	0.9
<i>previdência</i>	0.5
<i>senado</i>	0.6



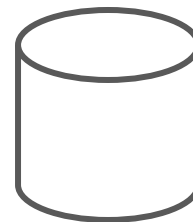
Pageview



Aurelio



Perfis
de Usuários ($d-1$)



Perfis
de Usuários $d-0$



Usuário

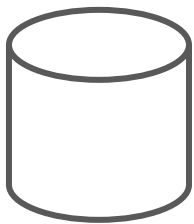
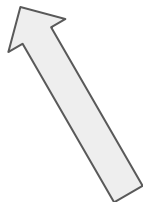
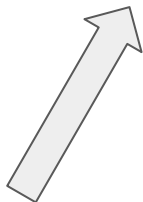


Documento	Score
<i>doc1</i>	0.9
<i>doc2</i>	0.7
<i>doc3</i>	0.65

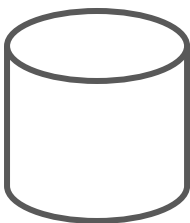
```
usersAndDocs = userProfiles (x) aurelio
```

```
userDocsScores = usersAndDocs.map { (user, doc) =>  
  score = similarity(user.terms, doc.terms)  
  (user, doc, score)  
}
```

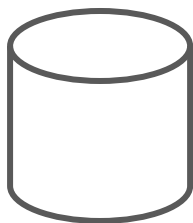
```
userDocsScores.groupByKey('user')
```



Documentos



Aurelio



Perfis
de Usuários (*d-0*)

10% ↑

CTR com Teste A/B

Content-Based Aurelio vs Content-Based Lucene

2% ↑

CTR com Teste A/B

Content-Based vs Filtragem Colaborativa

20% ↑

Taxa de Conversão por Algoritmo

Content-Based vs Filtragem Colaborativa para usuários *heavy*

Resultados

- Aumento no CTR
- Menor tempo de execução
 - Geração de Perfis de Usuário: ~1h
 - Geração de novas Recomendações: ~1h
- Valor em recomendações *content-based*

Próximas Ações

Próximas Ações

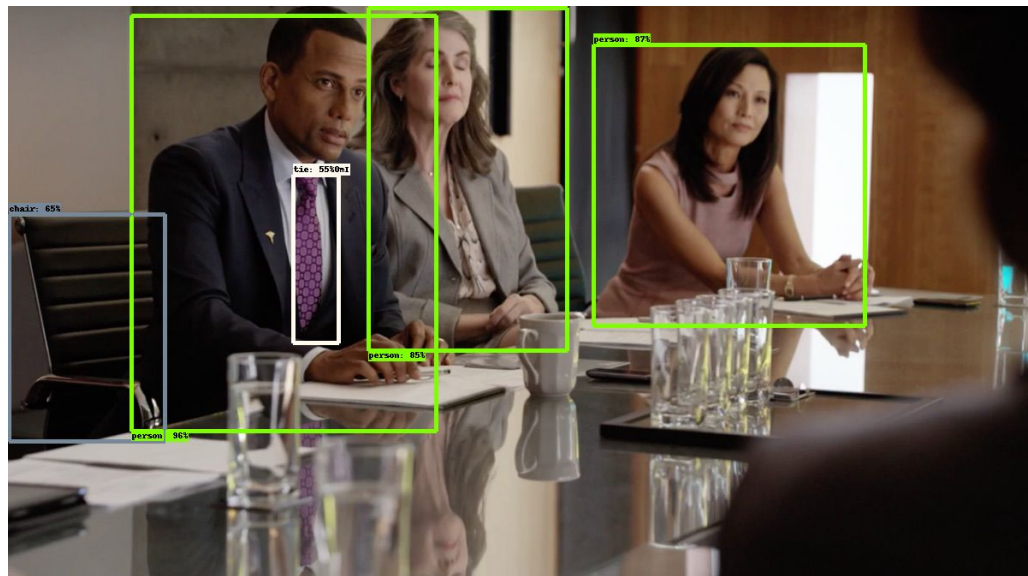
- Novas Fontes de Metadados
- Foco em **vídeo**
 - Reglobinition - Identificação de atores



Próximas Ações

- Novas Fontes de Metadados
- Foco em **vídeo**
 - Reglobinition - Identificação de atores
 - Extração de Features Visuais

```
{  
  "person": [0.96, 0.85, 0.87],  
  "tie": [0.55],  
  "chair": [0.65],  
  "giraffe": [0.42],  
  ...  
}
```



globo.com

We're Hiring!

<http://talentos.globo.com>