



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**INSTITUTO UNIVERSIDADE VIRTUAL**  
**CURSO DE SISTEMAS E MÍDIAS DIGITAIS**

**GUILHERME JUCÁ DE OLIVEIRA**

**UM SISTEMA PARA RECOMENDAÇÃO DE EVENTOS LOCAIS COM BASE NAS  
PREFERÊNCIAS DO USUÁRIO**

**FORTALEZA**

**2018**

GUILHERME JUCÁ DE OLIVEIRA

UM SISTEMA PARA RECOMENDAÇÃO DE EVENTOS LOCAIS COM BASE NAS  
PREFERÊNCIAS DO USUÁRIO

Relatório Técnico apresentado ao Curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito parcial para a obtenção do título de Bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Dr. Leonardo Oliveira Moreira

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

Guilherme, Jucá de Oliveira.

Um sistema para recomendação de eventos locais com base nas preferências do usuário. / Jucá de Oliveira Guilherme. – 2018.

51 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2018.

Orientação: Prof. Dr. Leonardo Oliveira Moreira.

1. Mídias Digitais. I. Título.

CDD 302.23

---

GUILHERME JUCÁ DE OLIVEIRA

UM SISTEMA PARA RECOMENDAÇÃO DE EVENTOS LOCAIS COM BASE NAS  
PREFERÊNCIAS DO USUÁRIO

Relatório Técnico apresentado ao Curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito parcial para a obtenção do título de Bacharel em Sistemas e Mídias Digitais.

Aprovada em: \_\_\_/\_\_\_/\_\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Leonardo Oliveira Moreira (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Wellington Wagner Ferreira Sarmiento  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Emanuel Ferreira Coutinho  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Agradeço inicialmente à minha família, meu pai Márcia e Henrique, que me deram todo o apoio que eu precisei durante essa jornada, meu irmão Luiz Henrique e Marília que me deram, além de encorajamento, novas visões e perspectivas.

Agradeço aos meus colegas e professores do Sistemas e Mídias Digitais que me auxiliaram e guiaram durante minha formação acadêmica, me proporcionando experiências, idéias e um crescimento que eu jamais teria tido de outra forma. Destaque para Emília Fortes, Enéas Arrais e Emilio Thalles que ajudaram na concepção deste trabalho quando ele ainda era apenas um projeto de disciplina.

Agradeço especialmente ao Prof. Leonardo Moreira pela incrível orientação que recebi durante esse trabalho, por todos os momentos em que tudo que eu precisava era de um pouco de calma para que eu pudesse enxergar os problemas e resolvê-los e por todas as palavras de encorajamento.

## RESUMO

Sistemas de recomendação fazem, cada vez mais, parte da vida cotidiana das pessoas. Seja recomendando músicas em aplicativos como Spotify ou séries em serviços como Netflix, essas tecnologias vêm para fazer com que um produto atinja o maior número de pessoas que possa se interessar por ele. Eventos de grande popularidade como grandes *shows* e festivais têm uma facilidade maior de atingir seu público pela repercussão que possuem nas grandes mídias, porém pequenos eventos possuem poucas ferramentas para tal, tendo que se valer do “boca a boca” para alcançar seu público. Este relatório técnico tem como objetivo apresentar o desenvolvimento de um sistema para que as pessoas possam encontrar eventos de seu interesse. Esse sistema alia as técnicas de sistemas de recomendação e as tecnologias *web* para criar uma plataforma onde os usuários terão muito mais facilidade de encontrar eventos e aproveitar experiências que, talvez, nem soubessem que existiam. O trabalho foi dividido em três etapas, a primeira teve caráter exploratório para conhecer melhor o cenário de eventos e os hábitos dos usuários. A segunda teve caráter descritivo apresentando o projeto e desenvolvimento do sistema dando destaque ao processo. A terceira e última etapa consiste em avaliar a recomendação, considerada o aspecto principal do sistema.

## **ABSTRACT**

Recommendation systems are, more than ever, a part of people's lives. Be it by suggesting music in apps like Spotify or television series on Netflix, these technologies are there to help a product or service reach as many interested people as possible. Big events like shows by famous artists or great festivals have an easier time finding their target audience because of the coverage they receive from large media outlets. Small events, on the other hand, don't have as many tools and end up relying on word of mouth to reach their audience. This technical report intends to present the development of a system that will allow people to find events of their interest. Such system uses techniques from recommendation systems and web technologies to create a platform where people will be able to find events around them with greater ease and enjoy experiences that, perhaps, they wouldn't even know existed. The work has been divided in three phases, the first is meant to explore the event promotion scene and understand better the habits of the users. The second phase describes the project and development of the system highlighting the process. The third and last phase consists of evaluating the recommendation functionality, considered the most important in the system.

## LISTA DE FIGURAS

Figura 1 - Posicionamento dos Centros .....	13
Figura 2 - Designação de grupos aos dados .....	14
Figura 3 - Reposicionamento dos centros .....	14
Figura 4 - Repetição dos passos 2 e 3 até convergência .....	15
Figura 5 - Convergência do algoritmo após 14 iterações .....	17
Figura 6 - Fluxograma do algoritmo <i>K-means</i> .....	17
Figura 7 - Chamada HTTP contendo um documento XML para um <i>web service</i> RPC .....	19
Figura 8 - Dado em JSON .....	21
Figura 9 - Dado em XML .....	21
Figura 10 - Interesse em descobrir novos eventos (nenhum interesse a muito interesse) .....	24
Figura 11 - Critérios importantes para a participação em um evento .....	25
Figura 12 - Classes de entidade e acesso .....	27
Figura 13 - Classes de implementação <i>K-Means</i> .....	28
Figura 14 - Classes de implementação do <i>web service</i> .....	29
Figura 15 - Trecho de código da implementação dos <i>web services</i> .....	29
Figura 16 - Diagrama relacional .....	30
Figura 17 - Vetor de preferências do usuário .....	31
Figura 18 - Matriz de palavras-chave dos eventos .....	31
Figura 19 - Evento antes e após o processamento .....	32
Figura 20 - Resultados inicial e final do usuário 1 .....	37
Figura 21 - Resultados inicial e final do usuário 4 .....	38
Figura 22 - Resultados inicial e final do usuário 5 .....	39
Figura 23 - Resultados inicial e final do usuário 3 .....	40
Figura 24 - Resultados inicial e final do usuário 7 .....	41

## LISTA DE TABELAS

Tabela 1 - Resultados do teste de desempenho .....	35
--	----

## LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
IBGE	Instituto Brasileiro de Geografia e Estatística
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
MVC	Model-View-Controller
HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
REST	Representational State Transfer
RPC	Remote Procedure Call
XML	Extensible Markup Language
JSON	JavaScript Object Notation
SGBD	Sistema de Gerenciamento de Bancos de Dados
DML	Data Manipulation Language
DDL	Data Definition Language
SQL	Structured Query Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	10
<b>1.1</b>	<b>Objetivos</b>	11
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	12
<b>2.1</b>	<b>Análise de Aglomerados</b>	12
<b>2.1.1</b>	<i>Entendendo o K-means</i>	12
<b>2.2</b>	<b>Aplicações Web</b>	17
<b>2.3</b>	<b>Web Services</b>	18
<b>2.4</b>	<b>Banco de Dados Relacional</b>	21
<b>3</b>	<b>METODOLOGIA</b>	22
<b>4</b>	<b>CONHECENDO OS EVENTOS</b>	23
<b>5</b>	<b>O SISTEMA</b>	26
<b>6</b>	<b>ESTRUTURA</b>	26
<b>7</b>	<b>CÁLCULO DAS RECOMENDAÇÕES</b>	30
<b>8</b>	<b>ASPECTOS FUNCIONAIS</b>	33
<b>9</b>	<b>AVALIAÇÃO</b>	34
<b>9.1</b>	<b>Recomendações</b>	34
<b>9.2</b>	<b>Desempenho</b>	34
<b>10</b>	<b>RESULTADOS E DISCUSSÕES</b>	35
<b>11</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	42

## 1 INTRODUÇÃO

Fortaleza, segundo o IBGE<sup>1</sup>, pode ser considerada a quinta maior capital do país em população com cerca de 2,5 milhões de habitantes. Grande parte da população possuirá uma vasta variação de gostos e interesses. Para aqueles que trabalham na indústria do entretenimento, um dos aspectos mais importantes é garantir que o seu produto chegue ao máximo de pessoas que possa ter interesse nele.

Eventos de maior popularidade, como grandes *shows* e festivais, possuem uma certa facilidade de atingir seu público alvo devido a exposição nas grandes mídias como rádio e televisão. Mas e os eventos de pequeno e médio porte? Eventos como pequenas produções teatrais, feiras de artesanato e competições de esportes menos populares possuem poucas ferramentas, além das páginas na internet e do “boca a boca”, para alcançar seu público. Como consequência, usuários interessados nesses eventos tem grande dificuldade de encontrar novas formas de entretenimento na cidade e acabam perdendo essas oportunidades.

Segundo a ANATEL<sup>2</sup>, o Brasil possui hoje cerca de 235 milhões de linhas celulares ativas, uma densidade de 113 celulares para cada 100 habitantes. A grande proliferação dessa tecnologia assim como as facilidades trazidas por ela, terminam por moldar os hábitos da nossa sociedade. Estar presente nessas plataformas se torna essencial para que os mais diversos eventos possam encontrar o seu público alvo.

Para solucionar esses problemas foi desenvolvido um sistema que servirá como uma plataforma onde produtores poderão expor seus eventos e participantes poderão encontrar eventos de seu interesse com muito mais comodidade.

O participante poderá usar esse serviço de forma gratuita para encontrar eventos em Fortaleza que correspondam aos seus interesses, salvar esses eventos em uma agenda e ainda convidar outras pessoas para esse evento. Os produtores poderão usar esse serviço para promover seus eventos e atingir seu público mais facilmente. Este sistema foi projetado para ser o *backend* de uma aplicação móvel.

Para garantir que o serviço será cômodo para os usuários ele contará com um sistema de recomendação que, baseado no perfil do usuário, verificará todos os eventos da plataforma para recomendar os eventos mais compatíveis com esse perfil. Removendo assim, a necessidade de o que usuário explore uma grande lista de eventos para encontrar o que procura. A volatilidade do mercado

---

<sup>1</sup> Disponível em: <<https://www.ibge.gov.br/estatisticas-novoportal/sociais/populacao/2098-np-censo-demografico/9662-censo-demografico-2010.html?=&t=downloads>> Acesso em: 20 de Abril de 2018.

<sup>2</sup> Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/283-brasil-tem-236-2-milhoes-de-linhas-moveis-em-janeiro-de-2018>> Acesso em: 20 de Abril de 2018.

móvel faz com que os usuários abandonem qualquer aplicativo que não atenda suas necessidades instantaneamente. Por tanto o sistema deve ser capaz recomendar eventos relevantes já nos primeiros momentos de uso.

Sistemas de recomendação combinam várias técnicas computacionais e estatísticas para relacionar dados e gerar resultados a partir de um contexto e um objetivo. Esse tipo de sistema pode ser utilizado para comparar dados de várias coisas como filmes, fotos, livros, propagandas e produtos.

Esses sistemas têm ganhado grande relevância no ramo de tecnologia, sendo utilizados por grandes empresas como Facebook, Google, Netflix e Amazon para gerar uma grande vantagem competitiva graças à comodidade gerada para o usuário. O sistema usa dados das preferências do usuário e das características dos eventos para escolher que eventos recomendar.

## **1.1 Objetivos**

O objetivo principal deste trabalho é desenvolver um sistema capaz de recomendar eventos baseado nas preferências iniciais do usuário e de se adaptar a mudanças no perfil do usuário com o passar do tempo. Para isso será necessário cumprir os seguintes objetivos secundários:

- A. Estudar técnicas de criação de algoritmos de recomendação.
- B. Estudar tecnologias empregadas nas áreas de aplicações móveis e banco de dados.
- C. Aplicar essas técnicas na codificação de um sistema.
- D. Avaliar a eficácia e a eficiência do sistema em gerar resultados favoráveis aos usuários.

## 2 REFERENCIAL TEÓRICO

### 2.1 Análise de Aglomerados

Romesburg (2004, p. 4) define Análise de Aglomerados como uma série de técnicas matemáticas que podem ser usadas para classificar objetos baseados em características semelhantes.

Por exemplo, se pegarmos um grupo de pedras de um riacho, notarmos suas características físicas como tamanho, formato e cor, separando-os em seguida em pilhas baseado nas características similares, estaríamos realizando fisicamente uma análise de aglomerados. Cada pilha de pedras seria um aglomerado.

Esse tipo de análise vem ganhando cada vez mais força no mercado computacional devido a necessidade de trabalhar com grandes quantidades de dados e encontrar padrões nos mesmos. Hoje em dia é possível ver muitas aplicações dessas técnicas em diversas áreas:

- Análise e classificação populacional baseada em indicadores sociais (MARTINS, 2002).
- Análise e classificação de Universidades baseado em desempenho acadêmico (VALADKHANI; WORTHINGTON, 2006).
- Análise de padrões meteorológicos (LIU; GEORGE, 2005).

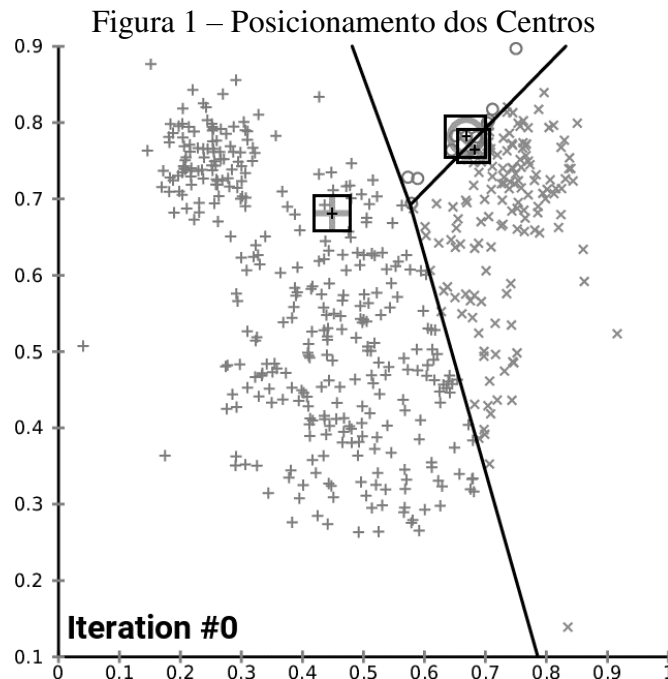
Existem vários algoritmos que utilizam as técnicas de aglomeração para gerar seus resultados, mas este trabalho terá foco nos algoritmos do tipo *K-means* por serem rápidos, eficientes e de propósito geral (TAN; STEINBACH; KARPATNE; KUMAR, 2018).

#### 2.1.1 Entendendo o *K-means*

Para realizar a classificação e o agrupamento de dados similares, o *K-means* segue os seguintes passos:

**1. Inicialmente são escolhidos  $K$  pontos arbitrários dentro da base de dados a ser analisada.**

Cada um desses pontos representará o centro de um grupo. A escolha inicial dos centros é determinada pelo tipo de resultado esperado. Por exemplo, se a intenção é encontrar padrões na base de dados, os centros serão escolhidos aleatoriamente. Porém, se a intenção é gerar grupos com base em padrões ou perfis previamente conhecidos nos dados, os centros devem ser escolhidos de acordo esses padrões. Vejamos um exemplo com  $K = 3$ .



Fonte: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

- 2. O segundo passo é designar que ponto da base de dados pertence a qual grupo.** Para isso é calculada a Distância Euclidiana entre o ponto em questão e os centros dos grupos. Quanto menor a distância calculada, maior a proximidade do ponto com o grupo. Esse processo ocorre para todos os dados a serem analisados. Distância Euclidiana ou distância entre pontos é uma função matemática comumente utilizada em cálculos geométricos. Tendo a posição de dois pontos em um gráfico bidimensional é possível calcular a distância entre eles. Para os pontos  $P = (P_x, P_y)$  e  $Q = (Q_x, Q_y)$  temos que a distância  $d$  é:

$$d = \sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2}$$

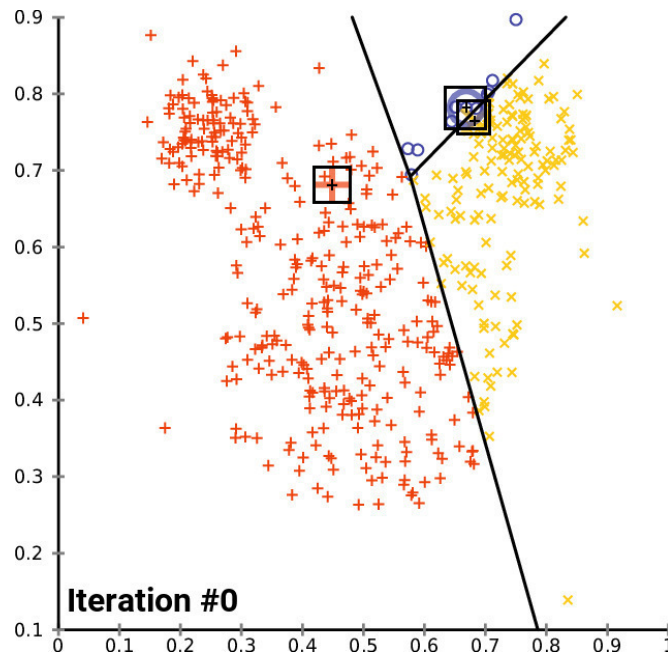
O que torna essa função útil para os cálculos em bases de dados é que ela pode ser computada para  $n$ -dimensões. Para os pontos  $P = (P_1, P_2, \dots, P_n)$  e  $Q = (Q_1, Q_2, \dots, Q_n)$  temos que a distância  $d$  é:

$$d = \sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \dots + (P_n - Q_n)^2}$$

$$\sqrt{\sum_{i=1}^n (P_i - Q_i)^2}$$

Isso significa que é possível encontrar a proximidade de objetos relacionando muitas características diferentes.

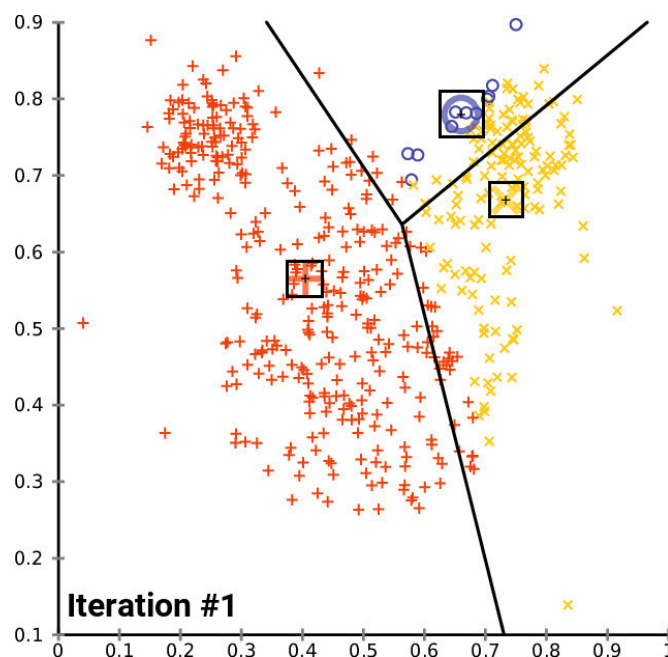
Figura 2 – Designação de grupos aos dados



Fonte: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

3. **O terceiro passo é re-calcular a posição dos centros dos grupos baseado nas distâncias médias entre todos os pontos do grupo.** Esse passo é vital pois a mudança do centro fará com que os cálculos de distância anteriores possam ser ajustados nos passos seguintes.

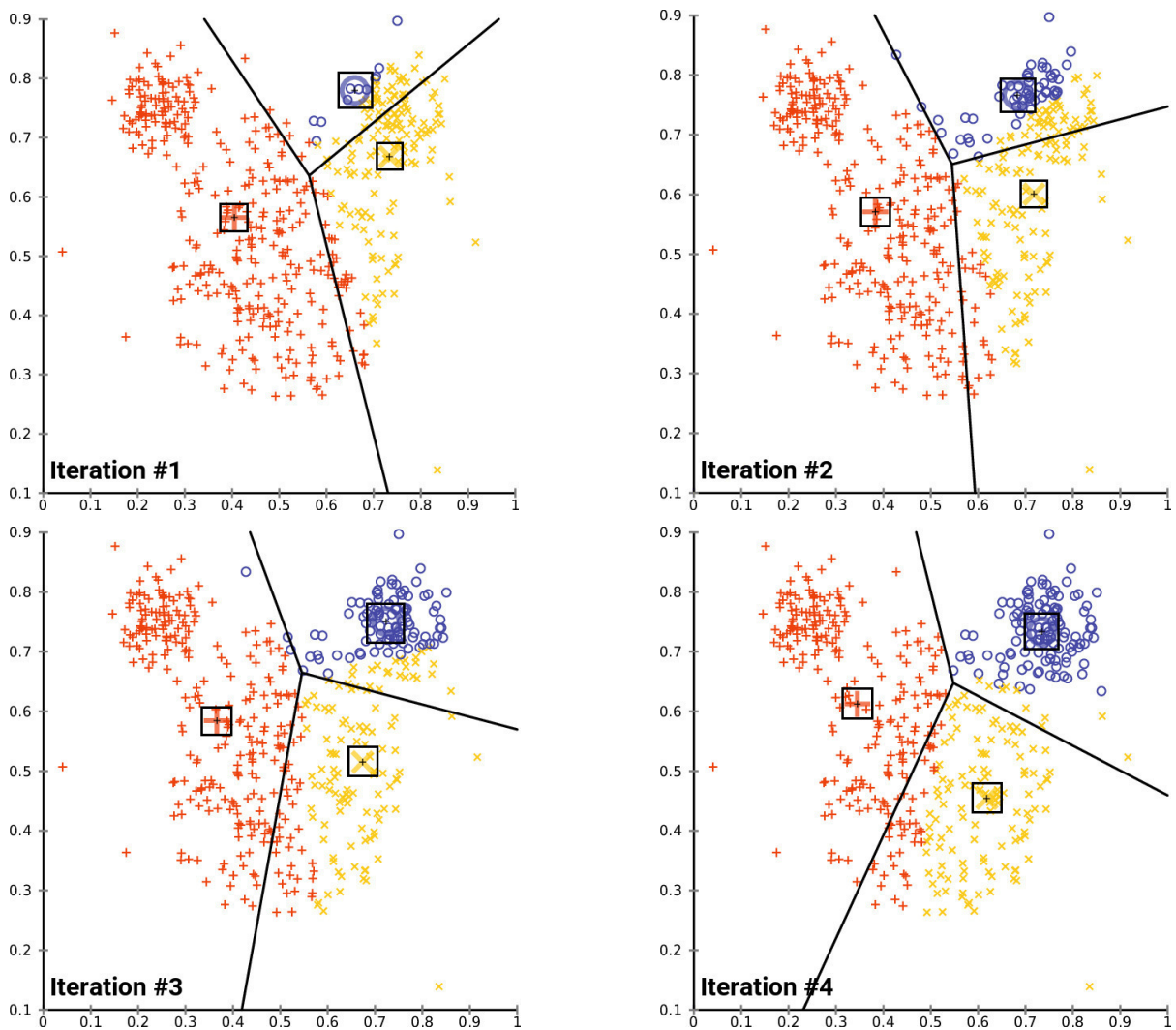
Figura 3 – Reposicionamento dos centros

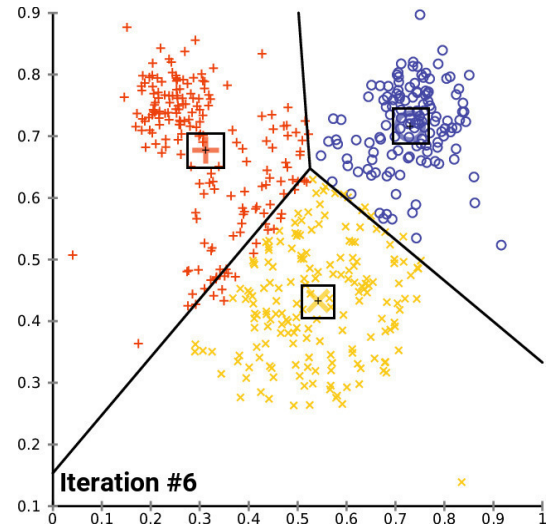
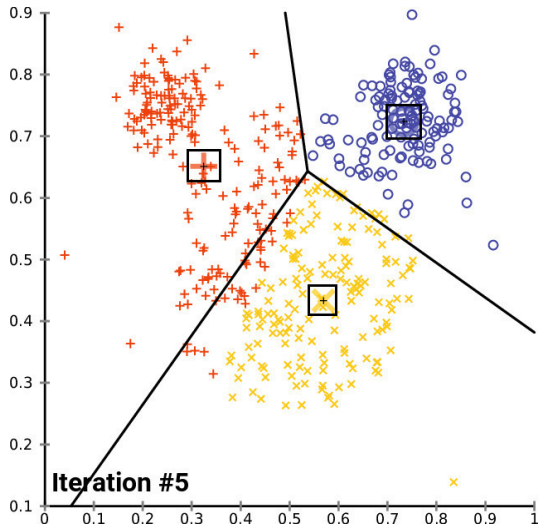


Fonte: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

- Os passos 2 e 3 se repetem até que não haja mudança significativa nas posições dos centros. Quando não há mais mudança nos centros significa que também não haverá mais mudança nos grupos.

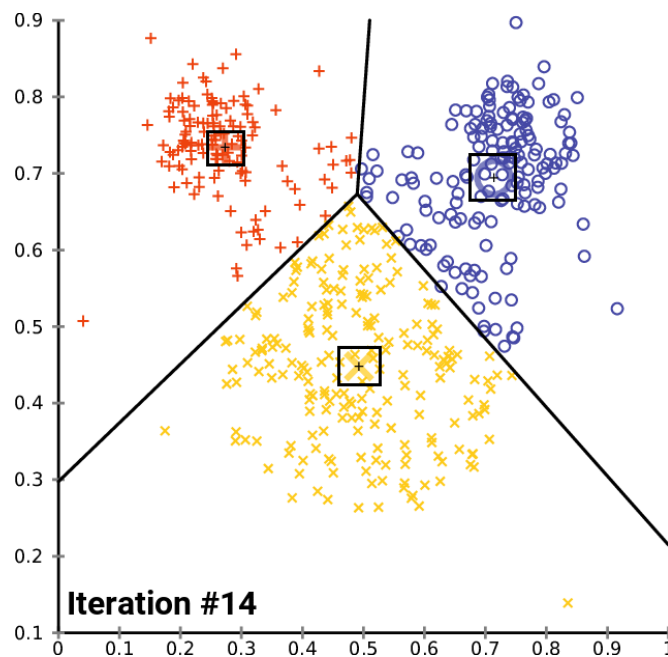
Figura 4 – Repetição dos passos 2 e 3 até convergência





Fonte: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

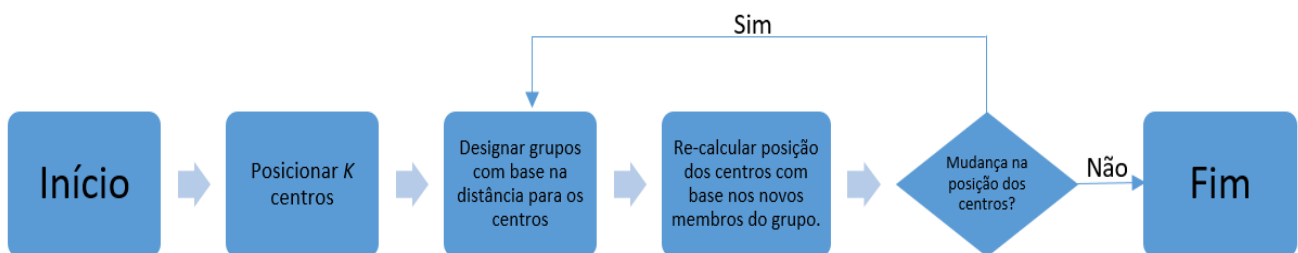
Figura 5 – Convergência do algoritmo após 14 iterações



Fonte: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering)

O exemplo mostrado nas Figuras 1 a 5 termina após 14 iterações e resulta em 3 grupos de objetos com características semelhantes dados os três centros escolhidos. Os passos do algoritmo se encontram resumidos na Figura 6.

Figura 6 – Fluxograma do algoritmo *K-means*



Fonte: elaborada pelo autor

## 2.2 Aplicações Web

Segundo Shklar e Rosen (2003) a internet ou “*World Wide Web*” foi proposta por Tim Berners-Lee como nada mais que uma biblioteca virtual, onde pesquisadores poderiam compartilhar informações e documentos através de endereços únicos.

Houve uma grande evolução desde então. Hoje em dia é possível não só compartilhar arquivos e documentos, mas realizar tarefas e cálculos complexos através de *websites* e especialmente de aplicações *web*.

Os autores definem uma aplicação *web* como uma aplicação cliente/servidor que utiliza um navegador *web* como cliente e realiza serviços interativos utilizando conexões com servidores através da internet.

Há várias arquiteturas para o desenvolvimento de aplicações *web* e uma das mais utilizadas é o *Model-View-Controller* (MVC). Gonçalves (2007) define a arquitetura MVC como um conceito (paradigma) de desenvolvimento que visa separar uma aplicação em três camadas:

- O Modelo, responsável pela representação dos dados da aplicação assim como todas as suas transformações. Em uma aplicação bancária, por exemplo, o Modelo é responsável por buscas na base de dados do banco e alterações nos valores.
- A Visão, responsável por mostrar os dados da aplicação ao usuário. Essa camada pode ser um *website*, uma aplicação móvel ou qualquer outra interface que permita o usuário interagir com a camada de Controle.
- O Controle, responsável pela comunicação entre as duas outras camadas. Em uma aplicação de comércio, por exemplo, o Controle recebe um pedido de produtos vindo da camada de Visão, interage com a camada de Modelo para obter os dados e retorna as informações pedidas.

### 2.3 Web Services

*Web Services* são aplicações *web* autocontidas, que podem ser acessadas remotamente utilizando protocolos *web* (Kalin, 2009). Devido a essa facilidade de acesso, os *web services* são capazes de transferir informações entre muitas tecnologias diferentes e muitas vezes incompatíveis. *Web services* são capazes de vários tipos de tarefas que vão de simples busca e entrega de informações até processos complexos e regras de negócio inteiras.

Com *web services* é possível criar todas as regras de negócio, operações complexas e processos de busca e entrega de dados codificando apenas uma vez. Pode-se depois consumir esse *web service* em várias aplicações diferentes como *websites* e aplicativos móveis.

Essa tecnologia é comumente utilizada para relações B2B (*Business to Business*) onde, por exemplo, uma aplicação que mostra promoções de supermercado em tempo real pode, utilizando o *web service* disponibilizado pelo Google Maps, calcular distâncias entre o usuário e os estabelecimentos facilmente.

Há várias arquiteturas para a criação de um *web service*, dentre elas destacam-se duas RESTful e RPC.

Antes que se possa entender os conceitos por trás dessas duas arquiteturas é necessário explicitar os conceitos de Método e Escopo quando se trata de *web services*:

- Método - Informação que, quando passada ao servidor, define que operação deve ser feita com os dados. Por exemplo, os métodos GET e DELETE presentes no protocolo HTTP dizem ao *web service* que ele deve enviar dados ao cliente ou apagar os dados respectivamente.
- Escopo - Informação que, quando passada ao servidor, define quais dados serão utilizados para realizar as operações. Seguindo o exemplo anterior, o escopo diria ao *web service* que dados devem ser entregues ou apagados respectivamente.

Segundo Richardson e Ruby (2007), a grande diferença entre as duas arquiteturas está em como elas utilizam o Método e o Escopo. A arquitetura RESTful recebe a informação do Método pelo protocolo HTTP e a informação do Escopo pela URI da chamada. Já a arquitetura RPC requer que o cliente envie um pacote de informações geralmente baseado no formato XML onde são definidas as informações do Método e do Escopo.

Uma chamada HTTP para o *web service* RESTful “*The Movie Database*” seria algo parecido com:

```
GET https://api.themoviedb.org/3/movie/{movie_id}/images
```

A chamada já define que o *web service* deve retornar as imagens associadas com o filme “*movie\_id*”. Já uma chamada HTTP para um *web service* RPC que procura produtos por meio de um código universal de produtos (UPC) seria algo como a Figura 7:

Figura 7 – Chamada HTTP contendo um documento XML para um *web service* RPC

```
POST /rpc HTTP/1.1
Host: www.upcdatabase.com
User-Agent: XMLRPC::Client (Ruby 1.8.4)
Content-Type: text/xml; charset=utf-8
Content-Length: 158
Connection: keep-alive
```

```
<?xml version="1.0" ?>
<methodCall>
  <methodName>lookupUPC</methodName>
  <params>
    <param><value><string>001441000055</string></value></param>
  </params>
</methodCall>
```

Fonte: Richardson; Ruby (2007) p.41

O método “*lookupUPC*” deve estar contido dentro do documento XML assim como o valor de parâmetro “001441000055” e todo esse envelope deve ser enviado para o servidor.

Há um grande debate entre desenvolvedores sobre qual a melhor arquitetura. Para aplicações no ambiente *web* e móvel, a arquitetura RESTful se mostra mais adequada (Kumari, 2015) por ser mais rápida, mais simples e envolver menos transferência de dados. Já para aplicações em computadores pessoais que exigem um nível de segurança mais elevado, a arquitetura RPC seria recomendada.

Outro fator importante na escolha entre as duas arquiteturas é o formato da resposta recebida. Por não depender do formato XML a arquitetura RESTful é capaz de usar plenamente o formato JSON (*JavaScript Object Notation*) que traz as informações de maneira mais simplificada e compatível com os padrões modernos de desenvolvimento *web*. Observe dois exemplos dos mesmos dados nas Figuras 8 e 9:

Figura 8 – Dado em JSON

```
[3, "three"]
```

Fonte: Richardson; Ruby (2007) p.45

Figura 9 – Dado em XML

```
<value>
  <array>
    <data>
      <value><i4>3</i4></value>
      <value><string>three</string></value>
    </data>
  </array>
</value>
```

Fonte: Richardson; Ruby (2007) p.45

## 2.4 Bancos de Dados relacionais

Silberschatz (2006, p. 24) define modelos de dados como um conjunto de ferramentas para descrever dados, as relações entre eles, suas semânticas e suas restrições de consistência.

O modelo relacional usa um conjunto de tabelas para representar tanto os dados quanto as relações entre eles. Sua simplicidade conceitual levou ao seu amplo uso; hoje, a grande maioria dos produtos de banco de dados é baseada no modelo relacional.

Para facilitar o entendimento sobre os conteúdos de bancos de dados relacionais é necessário explicitar alguns termos:

Sistema de Gerenciamento de Banco de Dados (SGBD) - Aplicação responsável por gerir e acessar os bancos de dados.

- Linguagem de Definição de Dados (DDL) - Linguagem que define a estrutura de armazenamento e as relações entre os dados.
- Linguagem de Manipulação de Dados (DML) - Linguagem que permite a manipulação dos dados presentes na base, com ela é possível selecionar, excluir ou incluir novos dados.

- SQL - Uma das linguagens de consulta mais utilizada dentro de sistemas de bancos de dados. Pode ser usada tanto DDL como DML.
- Tabela - Representação comum dos dados em um banco relacional. Cada tabela possui registros de um tipo específico e cada coluna da tabela representa um atributo desse tipo.

O uso dessas tecnologias vem revolucionando a sociedade moderna e criando coisas que, até alguns anos atrás, eram apenas ficção. Aplicações *web* são capazes de utilizar as tecnologias de banco de dados para armazenar e processar informações, permitindo que usuários de qualquer lugar do mundo possam ter acesso a informação ou serviços remotamente. *Web services* permitem uma fácil transmissão e integração entre várias aplicações *web* diferentes, aumentando muito sua produtividade. Análises de aglomeração utilizam todas essas tecnologias para inferir padrões e comportamentos dentro de um determinado número de dados.

### 3 METODOLOGIA

Em primeiro momento a pesquisa possuiu um caráter exploratório o que, segundo Gil (2002), proporciona uma maior familiaridade com o problema. O objetivo nessa etapa era levantar algumas das funcionalidades que devem estar presentes no sistema, assim como entender melhor o cenário de eventos em Fortaleza.

Nessa etapa foi realizada uma busca por eventos através dos meios convencionais como televisão e jornais, e meios modernos como redes sociais e marketing digital. Foram realizadas também entrevistas com profissionais da área de organização e promoção de eventos para tentar entender como eles buscavam atingir seus respectivos públicos alvo. Tendo em mente um cliente móvel para o sistema, foi realizada também uma busca de soluções similares para levantar quais as funcionalidades seriam necessárias.

Algumas das funcionalidades esperadas são:

- Cadastro e atualização das informações dos participantes dos eventos.
- Cadastro e atualização das informações dos criadores de eventos.
- Criação dos eventos e armazenamento das informações dos mesmos no banco de dados da aplicação.
- Envio das informações requisitadas pela aplicação móvel de maneira remota por meio da internet.
- Processamento das informações do usuário para que as recomendações possam ser enviadas para a aplicação móvel.

- Armazenar quais eventos o participante demonstrou interesse e utilizar essas informações para melhorar as futuras recomendações.

A segunda fase da pesquisa teve um caráter descritivo tratando, assim, do desenvolvimento do sistema de recomendação. Foram modeladas as funcionalidade do algoritmo, decidindo como os dados serão tratados antes dos cálculos e quais informações serão mais relevantes para a recomendação. Nessa etapa também foi modelado o banco de dados, quais informações estarão contidas nele e quais as relações entre as entidades do banco. Nessa etapa também foram decididas que tecnologias seriam utilizadas para a produção do sistema, bem como as classes Modelo dos eventos, participantes e produtores, assim como as classes e métodos de Controle e acesso ao banco de dados que serão implantadas no servidor. Utilizou-se a linguagem Java 8 para o desenvolvimento do sistema de recomendação e a tecnologia de *web services* para realizar a comunicação desse sistema com a camada de Visão do aplicativo. O ambiente de desenvolvimento utilizado foi a NetBeans IDE 8.2 com servidor Glassfish e o SGBD foi feito utilizando PostgreSQL 9.6.

A última fase do trabalho tratou de avaliar a eficiência da aplicação em termos de desempenho do código e na sua capacidade de recomendar os eventos, tanto, para perfis de usuário complexos, quanto para perfis simples. Foi feita uma avaliação qualitativas utilizando dados simulados para as recomendações e uma avaliação quantitativa para o desempenho utilizando a ferramenta *Apache JMeter 5.0*<sup>3</sup>.

Para os propósitos deste teste foram criados eventos variados entre si e com características específicas que os tornam destinados a um determinado perfil de participante. Foram criados usuários com os perfis pouco definidos para simular participantes recém-chegados no aplicativo, e usuários com perfis bem definidos para simular participantes veteranos.

Por fim, um comparativo das duas avaliações foi feito para determinar se a eficiência do algoritmo foi satisfatória nas duas fases de uso do participante e como essa eficiência muda entre as duas fases.

#### **4 CONHECENDO OS EVENTOS**

O primeiro passo para a concepção do algoritmo de recomendação foi entender melhor os objetos a serem recomendados, assim como o público a quem se destinam. Foram realizadas entrevistas com profissionais da área de eventos e pesquisas em redes sociais e *websites* especializados para entender quais as práticas envolvidas na concepção e promoção de eventos. Para

---

<sup>3</sup> Disponível em: <<https://jmeter.apache.org/>> Acesso em: 30 de Novembro de 2018.

entender melhor o público alvo do sistema foi realizada uma pesquisa de mercado, utilizando um formulário *online* (Anexo 1).

As entrevistas com três profissionais revelaram que o uso de “personas” é muito comum durante a concepção e promoção dos eventos como forma de simular quem seriam os possíveis participantes do evento. Geralmente utilizando uma série de características para descrever não só dados demográficos sobre o público, mas também os seus gostos e preferências.

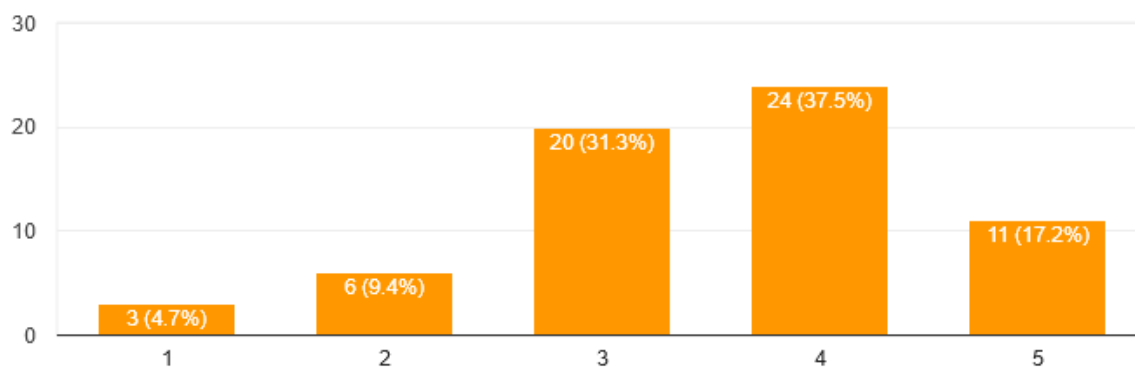
O objetivo do formulário era obter as seguintes informações sobre os futuros usuários da aplicação:

- Dados demográficos gerais como idade e escolaridade, para determinar o nível de familiaridade do público com as tecnologias móveis.
- Que tipos de eventos os usuários costumam ir.
- Com que frequência buscam eventos novos e que ferramentas utilizam para isso, para entender melhor os hábitos dos usuários.
- Se possuem interesse em buscar novos eventos e porque.
- Se consideram fácil encontrar eventos de seu interesse e porque.
- Se os usuários preferem ir aos eventos sozinhos ou acompanhados.
- Que critérios os usuários consideram determinantes para decidir se participarão ou não de um evento.

Figura 10 – Interesse em descobrir novos eventos (nenhum interesse a muito interesse)

### Como você avalia o seu interesse de descobrir novos eventos do seu gosto?

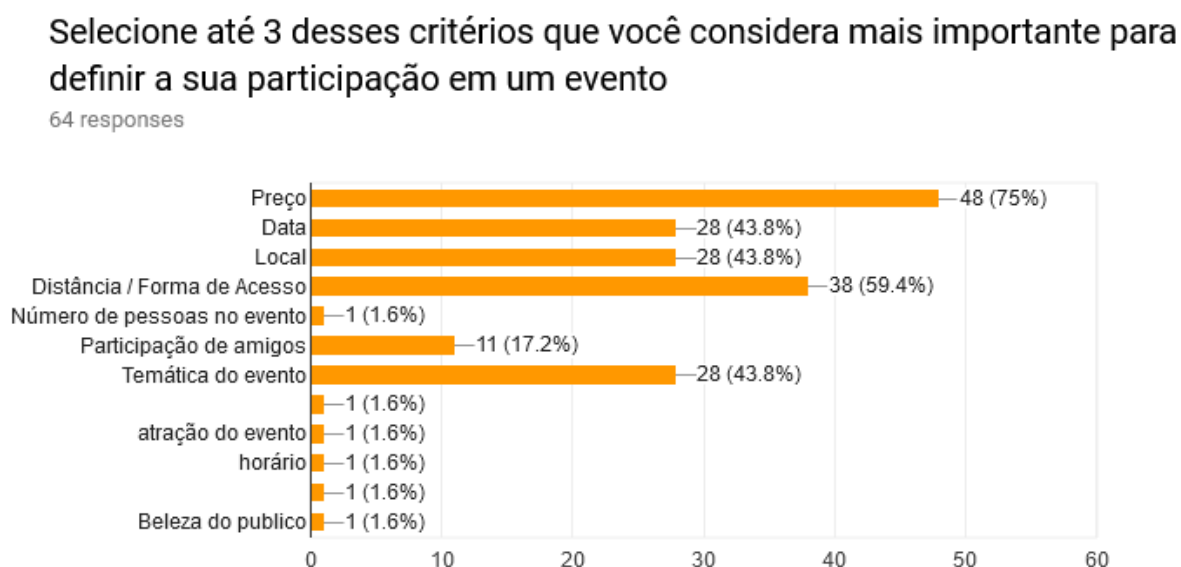
64 responses



Fonte: elaborada pelo autor

Com base no questionário foi possível determinar que há um interesse considerável em buscar novos eventos, em torno de 54% (Figura 10), porém os usuários encontram uma série de dificuldades na hora de buscar esses eventos. Falta de um local unificado para essa busca, falta de tempo para buscar os eventos ativamente, falta de confiança ou informação de onde procurar.

Figura 11 – Critérios importantes para a participação em um evento.



Fonte: elaborada pelo autor

Metade das respostas na Figura 11 indica que os usuários iriam a eventos mesmo que desacompanhados, então recursos que envolvem convidar outros usuários ou saber se seus amigos confirmaram presença em eventos específicos é importante, porém não urgente. As respostas também indicam que, apesar de importante, a temática do evento não é tão importante quanto o preço e a distância na tomada de decisão dos usuários. O participante pode ter interesse no evento, mas se ele for muito longe ou muito caro ele não irá participar.

Utilizando todas as informações obtidas foram tomadas as seguintes decisões sobre o sistema de recomendação:

- O algoritmo de recomendação deveria focar em recomendar levando em contas apenas as temáticas do evento. Os outros fatores como preço e distância, apesar de relevantes, iriam reduzir as possibilidades de descoberta de novos eventos dos usuários.

- Seria utilizado um conjunto de palavras-chave para descrever as preferências dos usuários, tornando assim mais fácil dos produtores de eventos identificarem mais facilmente que tipo de público eles querem atingir.
- O aplicativo deveria ter opções de exploração entre todos os eventos como filtros por data e preço, mas o foco principal seria na recomendação.

## 5 O SISTEMA

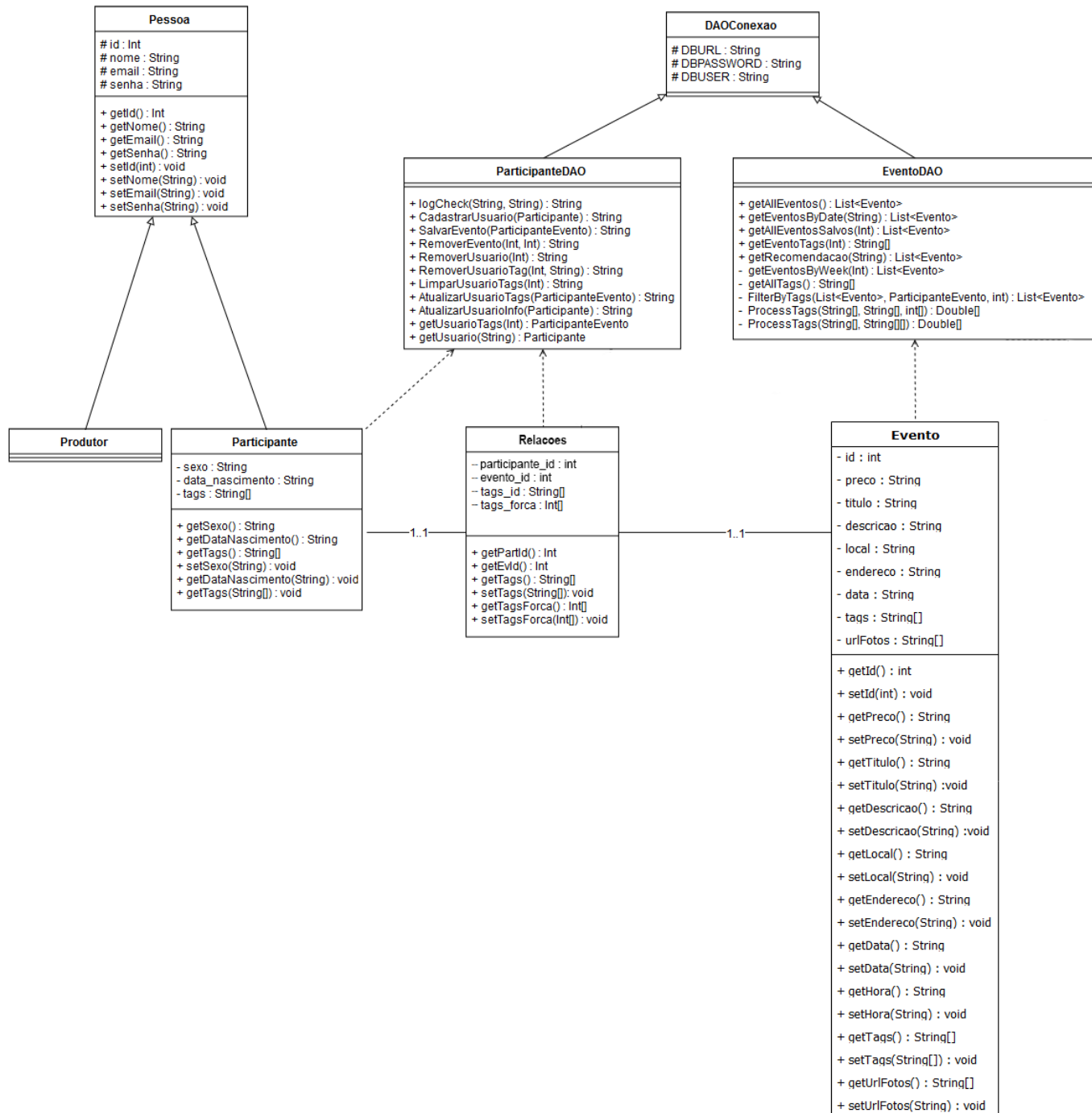
O sistema tem por objetivo servir de *backend* para uma aplicação capaz de ajudar pessoas a encontrar eventos de seus gostos pela cidade de Fortaleza. Ele foi desenvolvido seguindo o padrão *Model-View-Controller* visando especificamente as camadas de Modelo e Controle.

## 6 ESTRUTURA

O sistema pode ser dividido em três áreas principais: Acesso e manipulação das informações, cálculo das recomendações e comunicação com os fatores externos ao sistema.

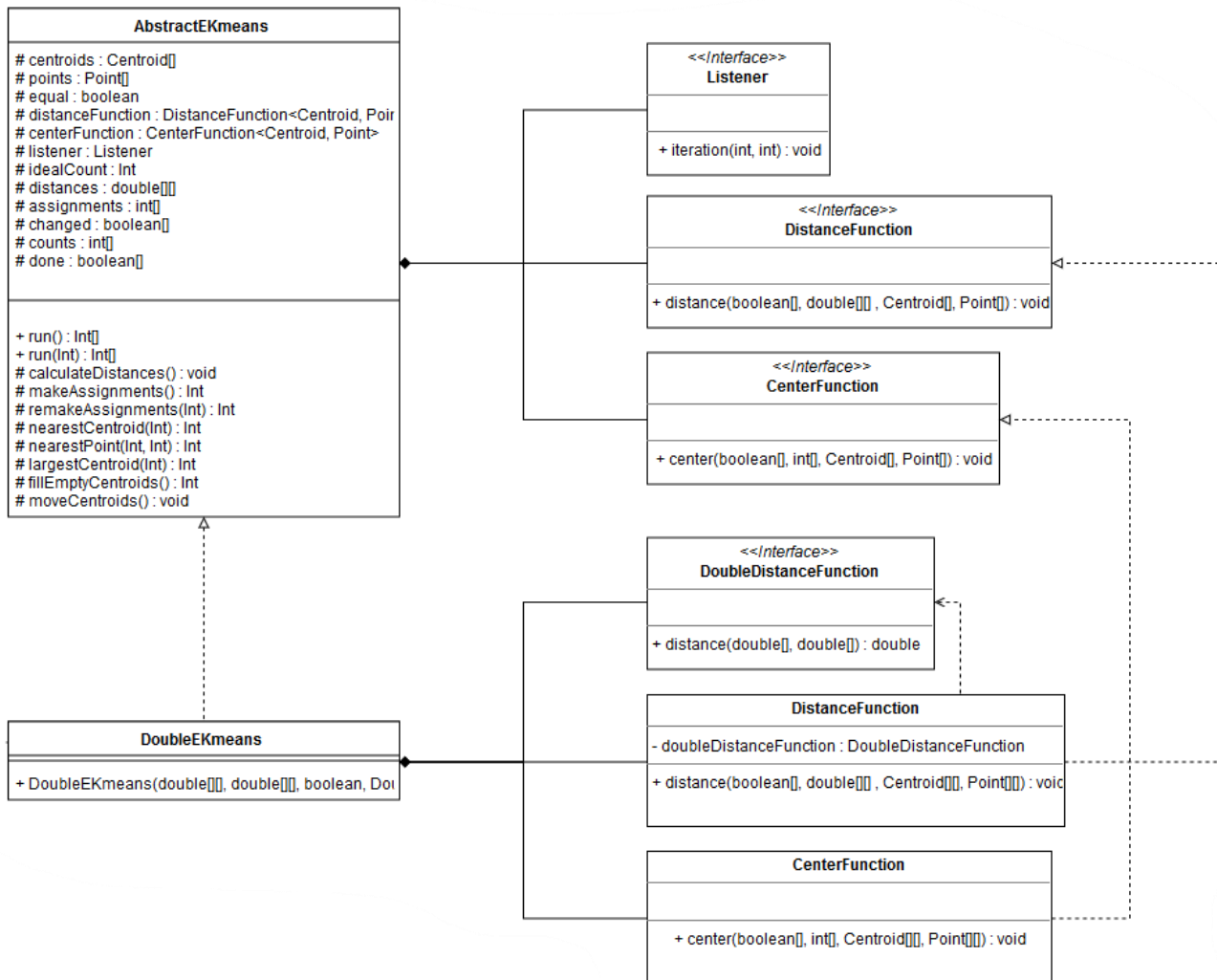
As informações contidas no banco de dados são representadas pelas classes *Pessoa*, *Participante*, *Produtor*, *Evento* e *Relacoes*, essa representação é necessária para que os dados possam ser manipulados. O acesso às informações é feito por meio das classes *ParticipanteDAO* e *EventoDAO*. Elas oferecem uma interface através da qual é possível inserir, atualizar e obter informações do banco de dados entre outras funcionalidades. Essas classe estão representadas na Figura 12.

Figura 12 – Classes de entidade e acesso.



Fonte: elaborada pelo autor

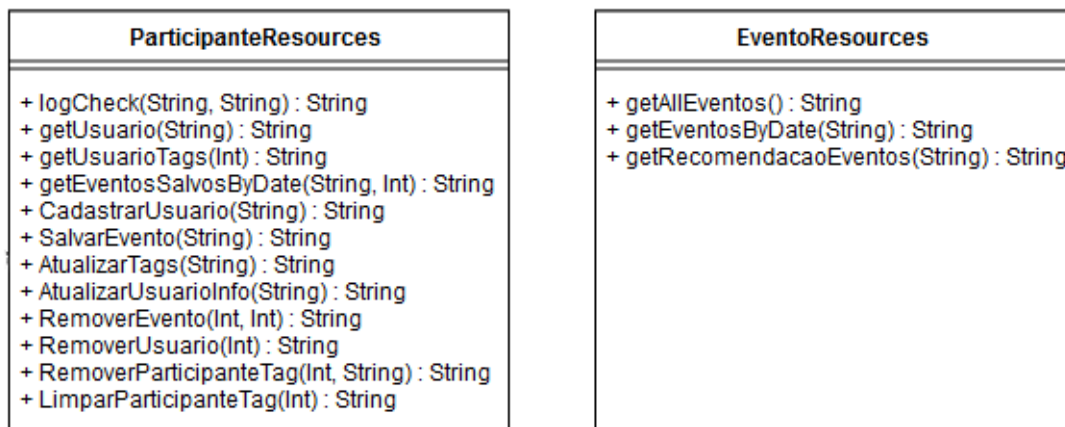
As classes responsáveis pelo cálculo da recomendação, demonstradas na Figura 13, são invocadas pela classe *EventoDAO* durante o cálculo das recomendações.

Figura 13 – Classes de implementação *K-Means*.

Fonte: elaborada pelo autor

As classes de comunicação *ParticipanteResources* e *EventoResources* implementam as tecnologias de *web service RESTful* e cada uma responde a uma chamada http e url e método HTTP específicos. As classes, representadas na Figura 14, utilizam o formato JSON para formatar as informações na chegada e na saída dos dados. O uso dessas tecnologia permite uma grande flexibilidade nas maneiras de consumir os serviços do sistema. A aplicação *frontend* pode ser um *website*, um aplicativo móvel ou apenas parte de uma aplicação maior. Um trecho de código dessa implementação se encontra na Figura 15.

Figura 14 – Classes de implementação do *web service*.



Fonte: elaborada pelo autor

Figura 15 – Trecho de código da implementação dos *web services*.

```

@GET
@Produces(MediaType.APPLICATION_JSON) //
@Path("GetUsuarioTags/{participante_id}")
public String getUsuarioTags(@PathParam("participante_id") int participante_id) {...7 linhas }

@GET
@Produces(MediaType.APPLICATION_JSON)
@Path("getAllEventosSalvos/{part_id}")
public String getEventosSalvosByDate(@PathParam("data") String data, @PathParam("part_id") int id) {...10 linhas }

/** Post method for Inserting data into database ...5 linhas */

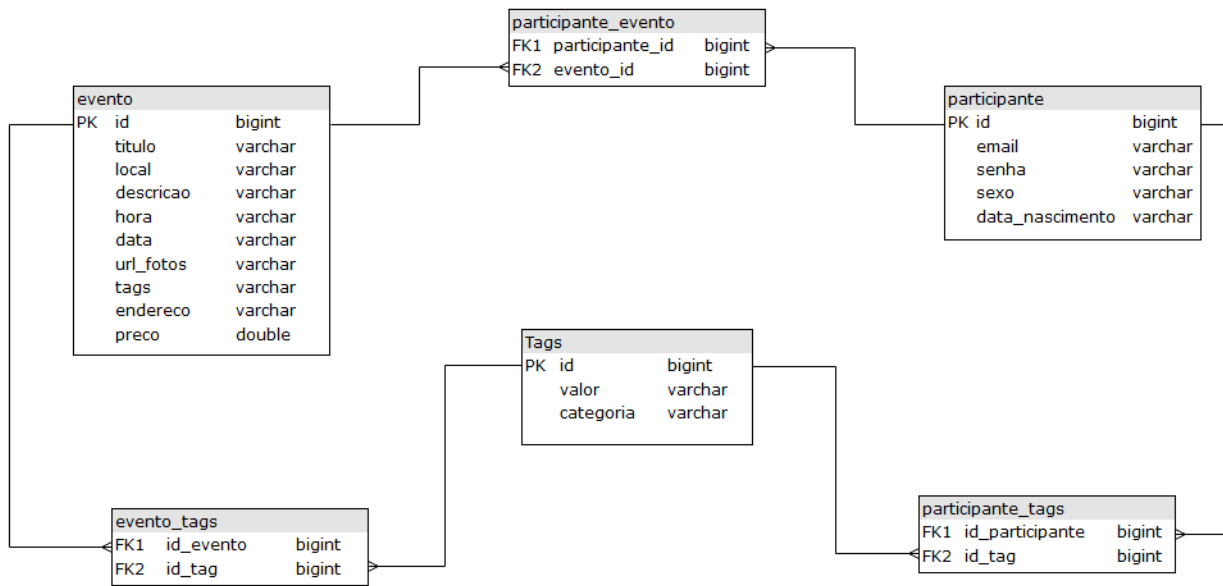
@POST
@Consumes({"application/json"})
@Path("CadastrarUsuario")
public String CadastrarUsuario(String content) {...15 linhas }

@POST
@Consumes("application/json")
@Path("SalvarEvento")
public String SalvarEvento(String content) {...10 linhas }
  
```

Fonte: elaborada pelo autor

O sistema utiliza um banco de dados relacional com a tabela *Tags* armazenando as palavras-chave utilizadas na recomendação e as demais armazenando as informações dos usuários e dos eventos e suas relações. As tabelas *evento\_tags*, *participante\_eventos* e *tags* não permitem repetição de linhas. O diagrama relacional do banco de dados encontra-se na Figura 16.

Figura 16 – Diagrama relacional.



Fonte: elaborada pelo autor

## 7 CÁLCULO DAS RECOMENDAÇÕES

O sistema de recomendação utiliza palavras-chave para mapear as características temáticas dos eventos em categorias de categorias. Novas palavras-chave e categorias podem ser adicionadas de acordo com a necessidade. Para garantir que as palavras-chave possuam uma boa significância dentro do sistema é necessário que elas sigam alguns critérios:

- Cada palavra-chave pertence apenas a uma categoria e é única dentro do sistema.
- Palavras-chave dentro de uma mesma categoria devem representar preferências diferentes dentro da mesma temática. Por exemplo: “reggae” e “rock” são palavras-chave da categoria Música, mas quem vai para um evento de “reggae” pode não querer ir a um evento de “rock”.
- Eventos possuem no máximo cinco palavras-chave associadas, para evitar saturação por parte do produtores de evento.
- As palavras chave não podem se repetir dentro do mesmo evento.
- Não há limites para o número de palavras chave associadas a um usuário, se o usuário possui a mesma palavra-chave associada várias vezes significa que ela é mais “forte” para ele.

Para calcular a recomendação utiliza-se a técnica de aglomeração *K-means* (TAN; STEINBACH; KARPATNE; KUMAR, 2018). São gerados 3 centros, de forma que, um desses centros representa as preferências do usuário. O algoritmo irá aglomerar os eventos com características semelhantes ao redor das preferências do usuário para gerar a lista de recomendação.

Para realizar as etapas do *K-means* é necessário processar as preferências do usuário e as palavras chave dos eventos para que todos tenham o mesmo formato e a distância entre os pontos possa ser calculada. Esse processamento é realizado dentro da classe *EventoDAO*, sendo a execução do *K-means* em si realizada através da classe *DoubleEKmeans*.

Primeiramente, é gerado um vetor de *strings* com todas as palavras-chave do sistema ordenadas por categoria (*allTagsArray*). Essa lista será utilizada como base para padronizar os eventos e as palavras-chave do usuário. Em seguida é gerada uma matriz de *strings* onde cada linha corresponde às palavras-chave associadas a cada evento (*eventosList*) e um vetor de *strings* contendo as palavras-chave associadas ao usuário (*userArray*). Encontram-se representações de *userArray* e *eventosList* nas Figuras 17 e 18 respectivamente.

Figura 17– Vetor de preferências do usuário.

[mpb, tatuagem, vidasaudavel, doacao, brecho, antiguidades, oficina, show, balada, carnaval, bebida, exercicio, jazz, pop, lual, rock, festival, aventura, moto, livros, reggae, poesia, literatura, social, samba, saude, exposicao, praia, indoor, calmo, culturabrasileira, gastronomia, kart, corrida, competicao]

Fonte: elaborada pelo autor

Figura 18 – Matriz de palavras-chave dos eventos.

[social, multirao, vidasaudavel, praia, meioambiente]  
 [basquete, social, indoor, rap, competicao]  
 [...]  
 [mpb, samba, social, indoor, culturabrasileira]  
 [oficina, calmo, literatura, culturabrasileira, poesia]

Fonte: elaborada pelo autor

Através do método *ProcessTags*, o *userArray* e as linhas de *eventosList* são transformados em vetores de *double* com tamanho igual a *allTagsArray*. Esse método vasculha *allTagsArray* e confere se aquela palavra chave está presente no vetor passado como parâmetro. Caso esteja ele adiciona o valor 1.0 à posição, caso contrário o valor fica em 0.0. O resultado final é que tanto as linhas de eventos quanto o vetor de usuário terão o mesmo tamanho e os valores



## 8 ASPECTOS FUNCIONAIS

Para que o sistema pudesse cumprir os requisitos levantados anteriormente foram implementadas várias funcionalidades além do cálculo da recomendação. Cada uma das funcionalidades deve ser acessada utilizando uma url especifica, o método HTTP adequado e enviar as informações corretas no corpo da chamada. Algumas das funcionalidades necessitam de informações específicas no corpo da chamada HTTP para serem executados de maneira correta, especialmente nas funcionalidades que envolvem cadastro e atualização de informações.

As funcionalidades relacionadas ao usuário incluem cadastro, atualização e remoção de usuários, *login* e manipulação das palavras-chave associadas ao seu perfil. Já as funcionalidades relacionadas aos eventos são todas voltadas para obtenção de informações sobre os mesmos. Alguns exemplos a seguir:

- Obter eventos recomendados
  - URL: `.../BeeThere/webresources/Evento/getRecomendacaoEventos/{email}`
  - Método: GET.
  - Entrada: nenhuma
  - Retorno: Objeto JSON contendo todos os eventos recomendados para o usuário especificado
- Desassociar um evento de um usuário
  - URL: `.../BeeThere/webresources/Participante/RemoverEvento/{id usuario}/{id evento}`
  - Método: DELETE.
  - Entrada: nenhuma
  - Retorno: Texto de confirmação
- Associar um evento a um usuário
  - URL: `.../BeeThere/webresources/Participante/SalvarEvento`
  - Método: POST.
  - Entrada: Objeto JSON contendo o id do usuário e o id do evento.
  - Retorno: Texto de confirmação.
- Atualizar dados do usuário
  - URL: `.../BeeThere/webresources/Participante/AtualizarUsuarioInfo`
  - Método: PUT.
  - Entrada: Objeto JSON contendo as informações do usuário modificadas.
  - Retorno: Texto de confirmação.

## 9 AVALIAÇÃO

A avaliação do sistema consiste em avaliar dois aspectos essenciais para garantir satisfação dos usuários com o sistema, a velocidade com que as recomendações são geradas e a precisão das mesmas. O cálculo da recomendação é de longe o processo mais intenso realizado, portanto, esse tipo de requisição é o maior suspeito em caso de baixa performance do sistema.

### 9.1 Recomendações

Para analisar a precisão das recomendações geradas pelo sistema foi realizada uma simulação onde foi criada uma base de dados contendo eventos e perfis de usuário com preferências variadas. Foi então previsto quais eventos deveriam ser recomendados para cada perfil de usuário e então os resultados das recomendações foram analisados.

Esse método foi escolhido por ser uma forma de se aproximar da técnica de criação de personas. Essa técnica é muito utilizada em várias áreas do design e da publicidade para tentar definir melhor o público alvo de um produto, serviço ou evento.

Os perfis de usuário pouco e muito definidos podem se diferenciar na quantidade e variedade de palavras-chave associadas a eles pois um usuário recém chegado no sistema terá poucas preferências salvas. É previsto que um perfil com uma grande variedade de preferências deverá receber recomendações proporcionalmente variadas. A lista com todos os perfis criados e suas palavras-chave estão em anexo (Anexo 2).

O comportamento previsto para o perfil do usuário é que à medida que ele vai associando eventos ao seu perfil, suas preferências vão sendo atualizadas. A tendência é que o perfil do usuário cresça primeiro lateralmente, aumentando a variedade de palavras-chave associadas, e depois passe a crescer mais verticalmente quando as preferências passam a se repetir e as palavras-chave ficam mais “fortes” no perfil do usuário.

### 9.2 Desempenho

Para avaliar o desempenho do sistema foi decidido que o fator mais importante a ser avaliado era o tempo de resposta da recomendação. Para isso foi utilizada a ferramenta Apache JMeter 5.0. Foram simuladas 5 rodadas de teste nas quais foram executados respectivamente 1, 10, 100, 500 e 1000 acessos simultâneos.

A máquina utilizada para realizar essa avaliação possui as seguintes características:

- CPU: Intel® Core™ i7-3770 3.40GHz
- Memória RAM: 16 GB RAM DDR3

- Disco: SATA HD 1TB
- Sistema Operacional: Windows 7 Professional 64 Bits

Em termos de software:

- Servidor: Glassfish Server 4.1.1
- Linguagem de Programação: Java 8
- Banco de Dados: Postgre SQL 9.6
- Software de avaliação: Apache JMeter 5.0

## 10 RESULTADO E DISCUSSÕES

O desempenho foi avaliado com base no tempo em que a aplicação respondia com as recomendações pedidas para um usuário, as imagens a seguir mostram os valores para 10, 25, 50, 75, 100, 125, 150, 300, 500, 750 e 1000 pedidos simultâneos.

Tabela 1 - Resultados do teste de desempenho.

Requisições Simultâneas	Valor Médio (ms)	Menor Valor (ms)	Maior Valor (ms)	Vazão	% de Erro
10	3081	4617	4617	2/sec	0
25	4612	1481	6918	3,2/sec	0
50	7996	1509	14581	3,2/sec	0
75	12005	1598	22332	3,2/sec	0
100	16122	1674	30691	3,2/sec	0
125	18853	1607	31735	3,8/sec	16
150	20713	1565	31135	4,7/sec	30
300	25651	1649	31880	9,3/sec	65
500	27804	1935	32080	15/sec	79,6
750	28716	1953	32311	22,5/sec	86,13
1000	29178	2203	32434	30,5/sec	89,6

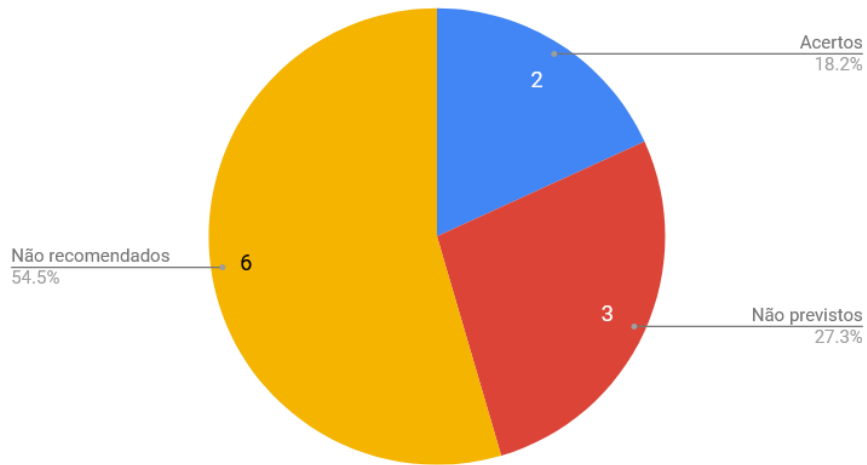
Foi possível observar que a partir de 125 requisições simultâneas a porcentagem de erro começa a aumentar, chegando a quase 90% ao final. Estima-se que isso ocorra pelo fato de estarem sendo utilizadas as configurações padrões do SGBD e do servidor, fazendo com que o servidor ou o SGBD neguem a requisição por falta de recurso computacional.

A precisão das recomendações teve de ser avaliada para cada usuário simulado, se um evento recomendado estivesse dentro da previsão ele era considerado um acerto.

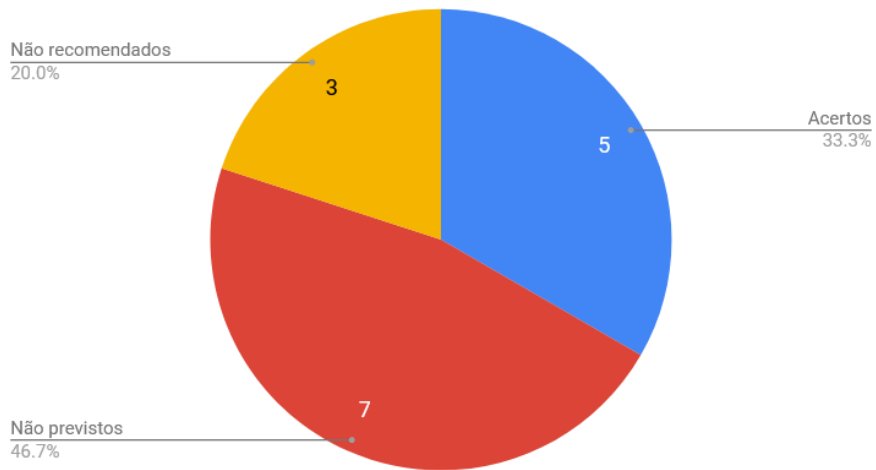
Os testes com os usuários 1, 2, 4, 5, 6, 8 e 9 tiveram comportamento dentro do esperado, com baixo número de eventos previstos ficando de fora da recomendação. O número de acertos aumentou com a adição de novas palavras-chave com exceção do usuário 6 cujo número de acerto foi o máximo já na recomendação inicial. O aumento do número de eventos não previstos também é um comportamento esperado dentro do sistema, pois o aumento na variedade de palavras-chave também aumenta a variedade nos eventos recomendados. Os resultados com os usuários 1, 4 e 5 estão representados nas Figuras 20, 21 e 22 respectivamente.

Figura 20 – Resultados inicial e final do usuário 1.

## Usuário 1 - Inicial



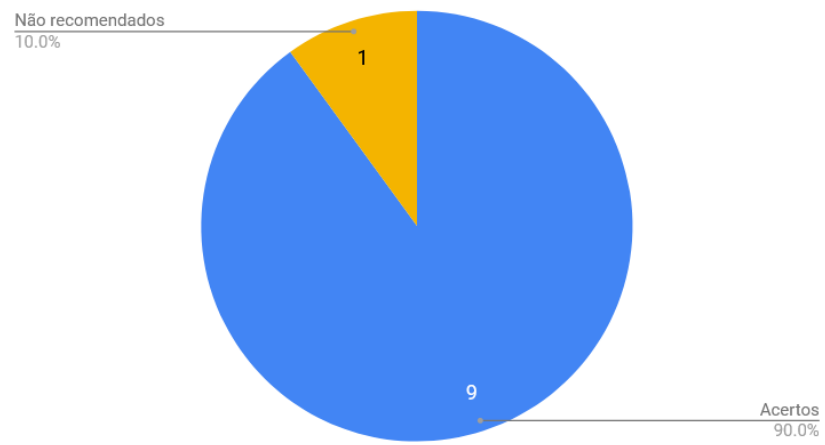
## Usuário 1 - Final



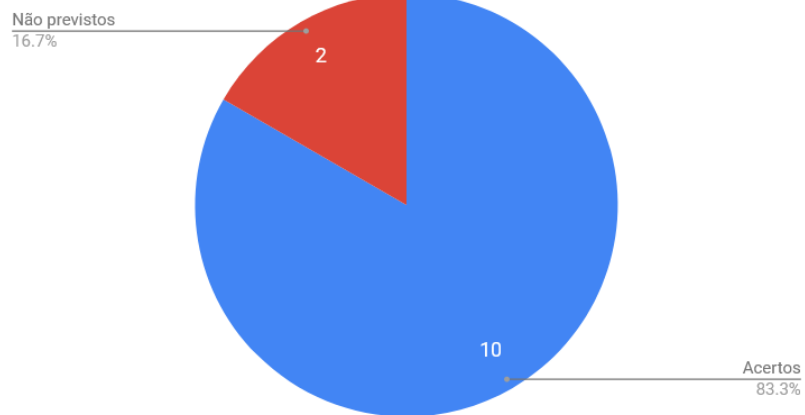
Fonte: elaborada pelo autor

Figura 21 – Resultados inicial e final do usuário 4.

## Usuário 4 - Inicial



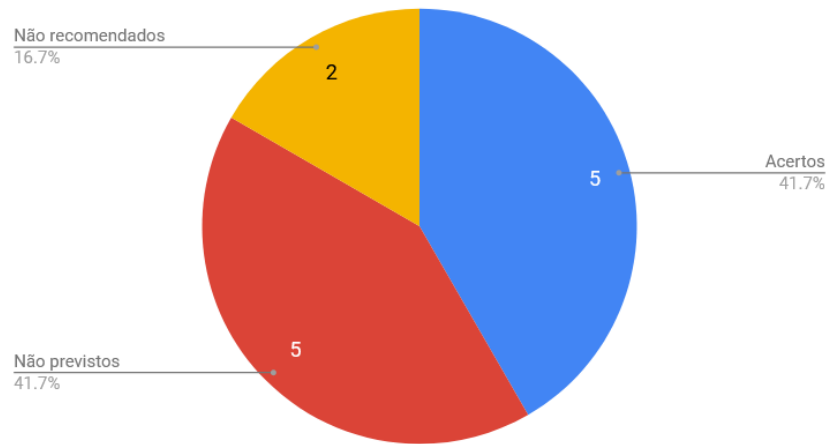
## Usuário 4 - Final



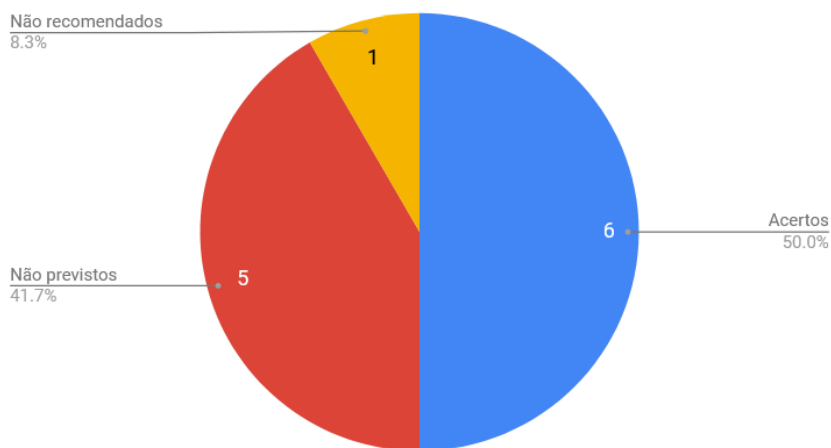
Fonte: elaborada pelo autor

Figura 22 – Resultados inicial e final do usuário 5.

## Usuário 5 - Inicial



## Usuário 5 - Final

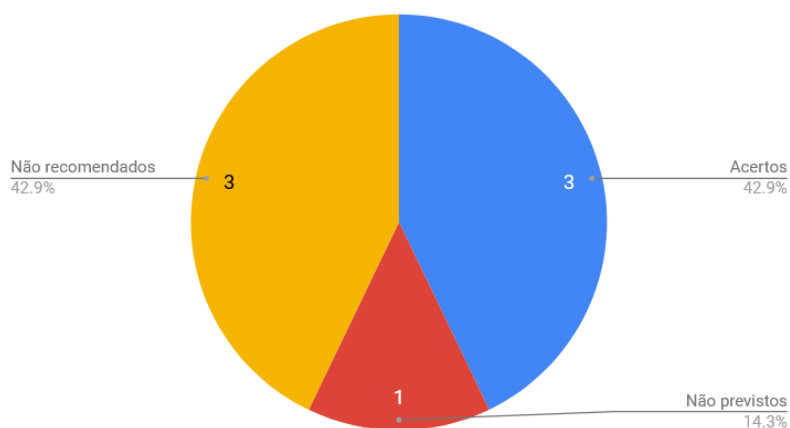


Fonte: elaborada pelo autor

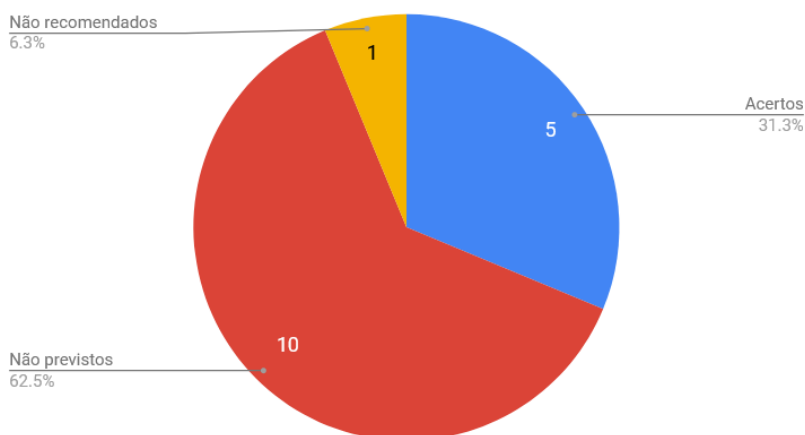
Os testes com os usuários 3 e 7 demonstraram um comportamento inesperado do sistema e merecem uma discussão mais aprofundada. no caso do usuário 3 (Figura 23) foi percebido que vários dos eventos não previstos que foram recomendados tinham muito pouco a ver com o perfil do usuário, com apenas uma palavra-chave em comum. Uma possível explicação para esse comportamento é a presença de palavras-chave pouco específicas ou não relacionadas diretamente a temática do evento como “social” e “indoor”. Apesar de quase todos os eventos previstos terem sido recomendados no final, houve uma aumento muito grande no número de eventos não previstos o que também pode ter sido causado pelo uso de palavras-chave com pouca especificidade.

Figura 23 – Resultados inicial e final do usuário 3.

## Usuário 3 - Inicial



## Usuário 3 - Final

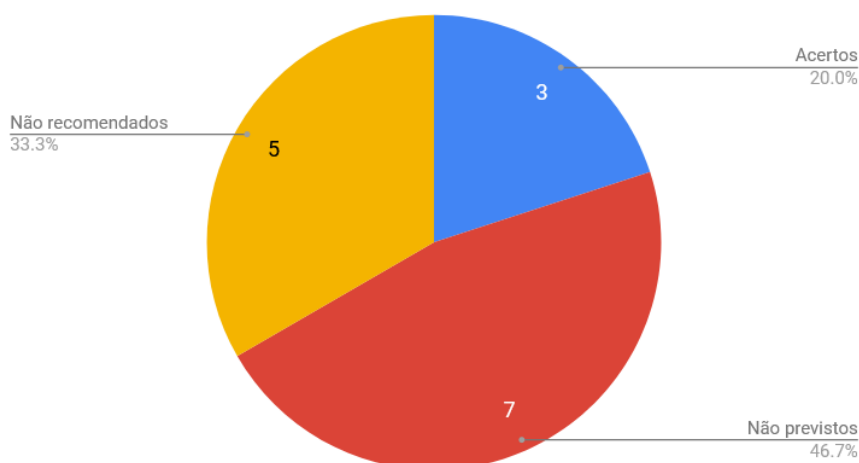


Fonte: elaborada pelo autor

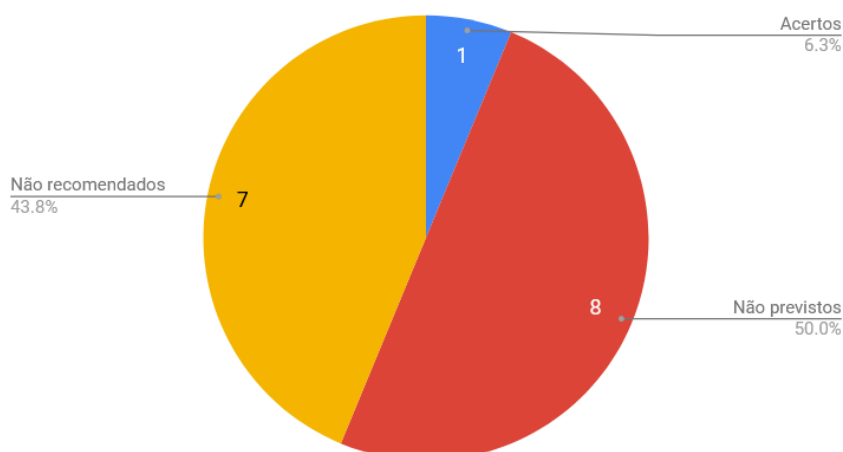
No caso do usuário 7 (Figura 24), o número de acertos diminuiu entre os pedidos e o percentual de acertos foi muito baixo. Isso pode indicar um erro no momento de prever que eventos seriam recomendados. Esse fator é importante pois o sistema espera que o produtor seja capaz de utilizar as palavras-chave corretas nos seus eventos para atingir o público correto. Se houverem muitos erros por parte dos produtores de evento o sistema se tornará muito ineficiente.

Figura 24 – Resultados inicial e final do usuário 7.

## Usuário 7 - Inicial



## Usuário 7 - Final



Fonte: elaborada pelo autor

Foi possível perceber durante a análise dos resultados que a presença de palavras-chave sem ligação direta com a temática do evento termina por atrapalhar a recomendação pois reduz o seu significado. palavras-chave que podem ser aplicadas a várias temáticas como “show”, “festiva”, “calmo” e “agitado” não podem ser usadas em conjunto com as outras, apesar de serem úteis na modelagem dos hábitos do usuário. Talvez um uso melhor delas seja na forma de filtros assim como o custo de entrada nos eventos e a distância.

Outra discussão que surge é se o sistema deveria maximizar o número de acertos possíveis, ou seja, mostrar o máximo de eventos que o usuário possa se interessar ou se ele deve evitar mostrar eventos não previstos, ou seja, evitar o erro o máximo possível em detrimento dos possíveis

acertos. Para responder essa pergunta seria necessário um teste com usuários reais para tentar definir melhor as expectativas deles sobre o funcionamento desse tipo de sistema.

## 11 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho mostrou o desenvolvimento de parte um sistema para a recomendação de eventos baseado nos gostos do usuário. Esse sistema deveria ser capaz de servir uma aplicação *frontend* com as regras de negócio necessárias ao seu funcionamento. Para isso foi realizada uma pesquisa exploratória Gil (2002) com o intuito de entender mais sobre a indústria de eventos e os hábitos dos usuários. Com o conhecimento adquirido foram decididas os requisitos necessários tanto para o sistema de recomendação quanto para as funcionalidades adicionais do sistema. Destacou-se o projeto e a estruturação do sistema, assim como os aspectos da sua implementação. Por fim, a funcionalidade de recomendação de eventos foi avaliada sob a perspectiva do desempenho de código e da precisão dos eventos recomendados.

Como trabalhos futuros pretende-se:

- A. Expandir o sistema para englobar temáticas de eventos ainda não foram abordadas, mas cuja variedade e complexidade exige atenção diferenciada como eventos infantis e LGBT;
- B. Criar funcionalidades para que os produtores de eventos possam adicionar novos eventos, além de obter informações sobre os eventos que criaram no sistema;
- C. Avaliar o desempenho do sistema com máquinas de qualidade de mercado para ter uma idéia melhor da sua capacidade;
- D. Realizar testes com usuários para avaliar melhor a precisão das recomendações e que tipo de expectativa eles têm em relação a esses sistemas;
- E. Implementar técnicas de segurança contra vulnerabilidades em sistemas *web* para proteger os dados dos usuários e produtores de eventos.
- F. Implementar técnicas de escalabilidade para aumentar o número de conexões simultâneas com um menor número de falhas.

## REFERÊNCIAS

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo Demográfico 2010**. Disponível em: <<https://www.ibge.gov.br/estatisticas-novoportal/sociais/populacao/2098-np-censo-demografico/9662-censo-demografico-2010.html?=&t=downloads>>. Acesso em: 20 de Abril de 2018.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Brasil tem redução de 574,38 mil linhas de telefonia móvel em fevereiro**. Disponível em: <<http://www.anatel.gov.br/dados/destaque-1/283-brasil-tem-236-2-milhoes-de-linhas-moveis-em-janeiro-de-2018>>. Acesso em: 20 de Abril de 2018.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4 ed. São Paulo: Editora Atlas, 2002.

GONÇALVES, Edson. **Desenvolvendo aplicações web com JSP, Servlet, Java Server Faces, Hibernate, EJB3 persistence, Ajax**. 1ª ed. Rio de Janeiro: Ciência Moderna, 2007.

MARTINS, Raul Aragão. **Uma tipologia de crianças e adolescentes em situação de rua baseada na análise de aglomerados (Cluster Analysis)**. *Psicol. Reflex. Crit.* [online]. 2002, vol.15, n.2, pp.251-260. ISSN 0102-7972. <http://dx.doi.org/10.1590/S0102-79722002000200003>.

KALIN, Martin. **Java Web Services: Up and Running**. Estados Unidos: O'Reilly, 2009.

RICHARDSON, Leonard; RUBY, Sam. **RESTful Web Services**. 4 ed. Estados Unidos: O'Reilly, 2007.

KUMARI, Vibha. **Web Services Protocol: SOAP vs REST**. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Volume 4 Issue 5, May 2015. Disponível em: < <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-4-ISSUE-5-2467-2469.pdf> >. Acesso em 10 de Maio de 2018.

ROMESBURG, H. Charles. **Cluster Analysis for Researchers**. Estados Unidos: Lulu Press, 2004.

EVERITT, Brian S.; LANDAU, Sabine; LEESE, Morven; STAHL, Daniel. **Cluster Analysis**. 5 ed. Reino Unido: Wiley, 2011.

LIU, Zhijian; GEORGE, Roy. **Mining Weather Data Using Fuzzy Cluster Analysis**. Berlin: Springer-Verlag, 2005.

VALADKHANI, A; WORTHINGTON, A. C. **Ranking and clustering Australian university research performance, 1998-2002**. Journal of Higher Education Policy and Management, 28(2), July 2006, 189-210. Taylor & Francis, 2006.

TAN, Pang-Ning; STEINBACH Michael; KARPATNE Anuj; KUMAR, Vipin. **Introduction to Data Mining**. 2 ed. Pearson, 2018.

WIKIPEDIA. **k-means clustering**. Disponível em: < [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering) > Acesso em: 7 de Maio de 2018.

APACHE JMETER. Disponível em: <<https://jmeter.apache.org/>> Acesso em: 30 de Novembro de 2018.

## APÊNDICES E ANEXOS

Anexo 1 - Formulário *online* da pesquisa sobre os hábitos dos usuários.

# Pesquisa para desenvolvimento de aplicativo

\*Required

## Dados demográficos

Preenche aqui pra gente te conhecer melhor. :)

Idade \*

Your answer

Escolaridade \*

Choose



Cidade \*

Your answer

Qual o sistema operacional do seu celular? \*

- Android
- iOS
- Windows Phone
- Não possuo smartphone
- Other: \_\_\_\_\_

Quanto tempo você utiliza o smartphone por dia? \*

- Menos de 1 hora
- Entre 1 e 3 horas
- Entre 4 e 8 horas
- Entre 9 e 12 horas
- Mais de 12 horas

### Quais os tipos de aplicativo que você utiliza diariamente \*

- Redes Sociais
- Filmes/Videos
- Transporte
- Mapas
- Entrega de comida
- Bancos
- Financeiros (para investimento ou controle de gastos)
- Lembrentes/Agendas
- Mercadorias/Vendas
- Encontros
- Musica
- Camera/Edição de imagem
- Jogos
- Educativos
- Other: \_\_\_\_\_

## Eventos de pequeno porte

Para nós, eventos de pequeno porte (ou de nicho), são eventos pouco divulgados pela grande mídia. Alguns exemplos desses tipos de evento:

- Feiras de Artesanato
- Eventos de jogos de tabuleiro
- Festivais de dança
- Partidas de Rugby

Você costuma participar de eventos de pequeno porte? \*

Sim

Não

## Participação em eventos

Fale mais de seus gostos :D

Quais os tipos de evento que você costuma ir? \*

Festivais Culturais

Apresentações

Encontros

Feiras

Competições

Exposições

Academicos

Oficinas

Other...

### Com que frequência você costuma buscar novos eventos? \*

- Costumo buscar novos eventos frequentemente (Semanalmente)
- Procuo eventos novos de vez em quando (Mensamente)
- Procuo eventos novos raramente, apenas quando algum me chama mais a atenção
- Não costumo procurar eventos novos, prefiro ir em eventos que eu já conheço

### Como você descobre eventos de seu interesse \*

Long-answer text

---

### Você costuma a ir acompanhado aos eventos? \*

- Gosto de ir sozinho
- Costumo ir sozinho mas prefiro ir acompanhado
- Costumo ir acompanhado mas às vezes vou sozinho também
- Só vou se for acompanhado
- Other...

Selecione até 3 desses critérios que você considera mais importante para definir a sua participação em um evento \*

- Preço
- Data
- Local
- Distância / Forma de Acesso
- Número de pessoas no evento
- Participação de amigos
- Temática do evento
- Other...

Como você avalia a facilidade de achar eventos de seu interesse? \*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Por que?

Long-answer text

---

Como você avalia o seu interesse de descobrir novos eventos do seu gosto? \*

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Por que?

Long-answer text

---

## Anexo 2 - Usuários simulados para a avaliação.

### Usuario Teste 1

- Palavras-chave Iniciais: poesia, teatro, culturabrasileira
- Palavras-chave Finais: mpb, samba, funk, danca, poesia, teatro, culturabrasileira, brecho, festival, fotografia.
- Eventos Previstos: 29, 35, 38, 39, 45, 51, 54, 57.
- Resultados Iniciais: 51, 17, 7, 38, 44
- Resultados Finais: 54, 51, 56, 55, 57, 9, 17, 21, 29, 36, 38, 44

### Usuario Teste 2

- Palavras-chave Iniciais: reggae, lual, parque
- Palavras-chave Finais: reggae, rap, surf, tatuagem, lual, praia, exploracao, meioambiente, parque, serra.
- Eventos Previstos: 2, 5, 16, 28, 40, 44, 49, 57, 27, 17
- Resultados Iniciais: 57, 40, 44
- Resultados Finais : 57, 17, 27, 28, 5, 44

### Usuario Teste 3

- Palavras-chave Iniciais: anime, quadrinhos, cosplay
- Palavras-chave Finais: anime, cosplay, quadrinhos, pop, eletronico, videogames, show, exposicao, fantasia.
- Eventos Previstos: 19, 22, 30, 43, 61, 1
- Resultados Iniciais: 8, 19, 1, 43
- Resultados Finais : 53, 59, 62, 61, 9, 12, 8, 13, 19, 25, 30, 1, 38, 22, 46

### Usuario Teste 4

- Palavras-chave Iniciais: filmes, boardgames, videogames
- Palavras-chave Finais: filmes, boardgames, videogames, cardgames, esports, quadrinhos, bebida, show, rpg, rock.
- Eventos Previstos: 1, 9, 12, 19, 22, 24, 30, 62, 63, 61
- Resultados Iniciais: 63, 61, 9, 12, 19, 24, 30, 1, 22
- Resultados Finais : 59, 63, 62, 61, 30, 12, 19, 25, 24, 30, 1, 22

### Usuario Teste 5

- Palavras-chave Iniciais: empreendedorismo, palestra, inovacao
- Palavras-chave Finais: empreendedorismo, palestra, tecnologia, inovacao, mesaredonda, tecnologiadainformacao
- Eventos Previstos: 11, 20, 36, 47, 48, 64, 8
- Resultados Iniciais: 50, 64, 48, 11, 20, 15, 39, 45, 47, 46

- Resultados Finais : 50, 64, 48, 11, 8, 20, 15, 39, 45, 47, 46

#### Usuario Teste 6

- Palavras-chave Iniciais: ensinossuperior, medicina, saude
- Palavras-chave Finais: ensinossuperior, minicurso, oficina, mesaredonda, biologia, saude, medicina
- Eventos Previstos: 15, 33, 60, 64
- Resultados Iniciais: 64, 60, 15, 33
- Resultados Finais : 64, 48, 20, 15, 47

#### Usuario Teste 7

- Palavras-chave Iniciais: exercicio, vidasaudavel, praia
- Palavras-chave Finais: exercicio, corrida, luta, rap, rock, forro, mpb, praia, show, vidasaudave.
- Eventos Previstos: 5, 6, 13, 16, 26, 41, 52, 56
- Resultados Iniciais: 49, 57, 60, 6, 16, 28, 33, 41, 40, 44
- Resultados Finais : 50, 57, 60, 13, 28, 33, 35, 40, 44

#### Usuario Teste 8

- Palavras-chave Iniciais: balada, bebida, funk
- Palavras-chave Finais: show, balada, carnaval, forro, pagode, bebida, samba, funk, festival
- Eventos Previstos: 2, 9, 18, 34, 37, 56, 57, 12
- Resultados Iniciais: 56, 9, 2, 12, 8, 18, 24, 37
- Resultados Finais : 54, 56, 55, 9, 21, 2, 12, 8, 10, 13, 18, 25, 24, 35, 37, 43

#### Usuario Teste 9

- Palavras-chave Iniciais: livros, jazz, culturabrasileira
- Palavras-chave Finais: literatura, pintura, antiguidades, jazz, samba, mpb, culturabrasileira, gastronomia, exposicao, livros.
- Eventos Previstos: 7, 10, 17, 23, 29, 35, 38, 39, 45, 54, 51, 55, 21, 55
- Resultados Iniciais: 51, 17, 29, 7, 23, 39, 38, 44
- Resultados Finais : 54, 51, 56, 55, 59, 57, 21, 12, 17, 24, 29, 30, 7, 23, 39, 38, 44, 46