



# BETA - FRAMEWORK PARA APLICAÇÃO DA TÉCNICA DE BEHAVIOR TARGETING

RUDIMAR IMHOF



# SUMÁRIO

- 1 INTRODUÇÃO
- 2 FUNDAMENTAÇÃO TEÓRICA
- 3 DESENVOLVIMENTO
- 4 CONCLUSÕES
- REFERÊNCIAS BIBLIOGRÁFICAS



# 1. INTRODUÇÃO

- WEB 3.0
- WEB ANALYTICS
- DATA MINING
- FRAMEWORK
- BETA



# 1.1 OBJETIVOS DO TRABALHO

- OBJETIVO GERAL
  - DESENVOLVER UM FRAMEWORK PARA ANÁLISE DAS INTERAÇÕES DO USUÁRIO EM UMA APLICAÇÃO WEB E FORNECER DADOS ANALÍTICOS DESSAS INTERAÇÕES UTILIZANDO A TÉCNICA DA SEGMENTAÇÃO COMPORTAMENTAL



# 1.1 OBJETIVOS DO TRABALHO

- OBJETIVOS ESPECÍFICOS:
  - SEGMENTAR OS DADOS DE ACORDO COM O USO DO SISTEMA;
  - APLICAR MÉTRICAS PARA ANÁLISE DE PERFIS;
  - FORNECER OS GRUPOS SEGMENTADOS AO QUAL O USUÁRIO DA APLICAÇÃO PERTENCE.



# 2 FUNDAMENTAÇÃO TEÓRICA

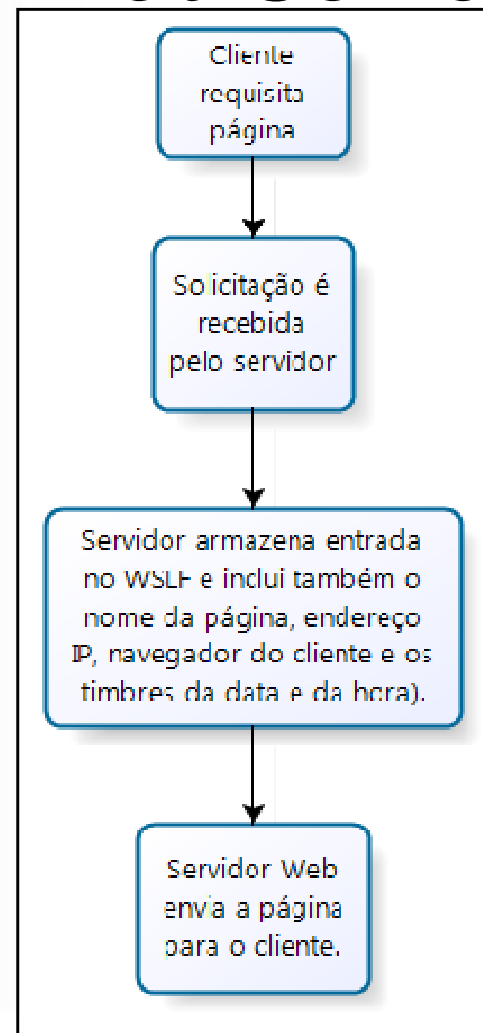
- 2.1 WEB ANALYTICS E BEHAVIOR TARGETING
- 2.2 DATA MINING
- 2.3 DESENVOLVIMENTO DE FRAMEWORKS
- 2.4 TRABALHOS CORRELATOS



## 2.1 WEB ANALYTICS E BEHAVIOR TARGETING

## 2.1.1 Fluxo de cliques

- web logs - Web Server Log Files







## 2.1.1 Fluxo de cliques

- web beacons

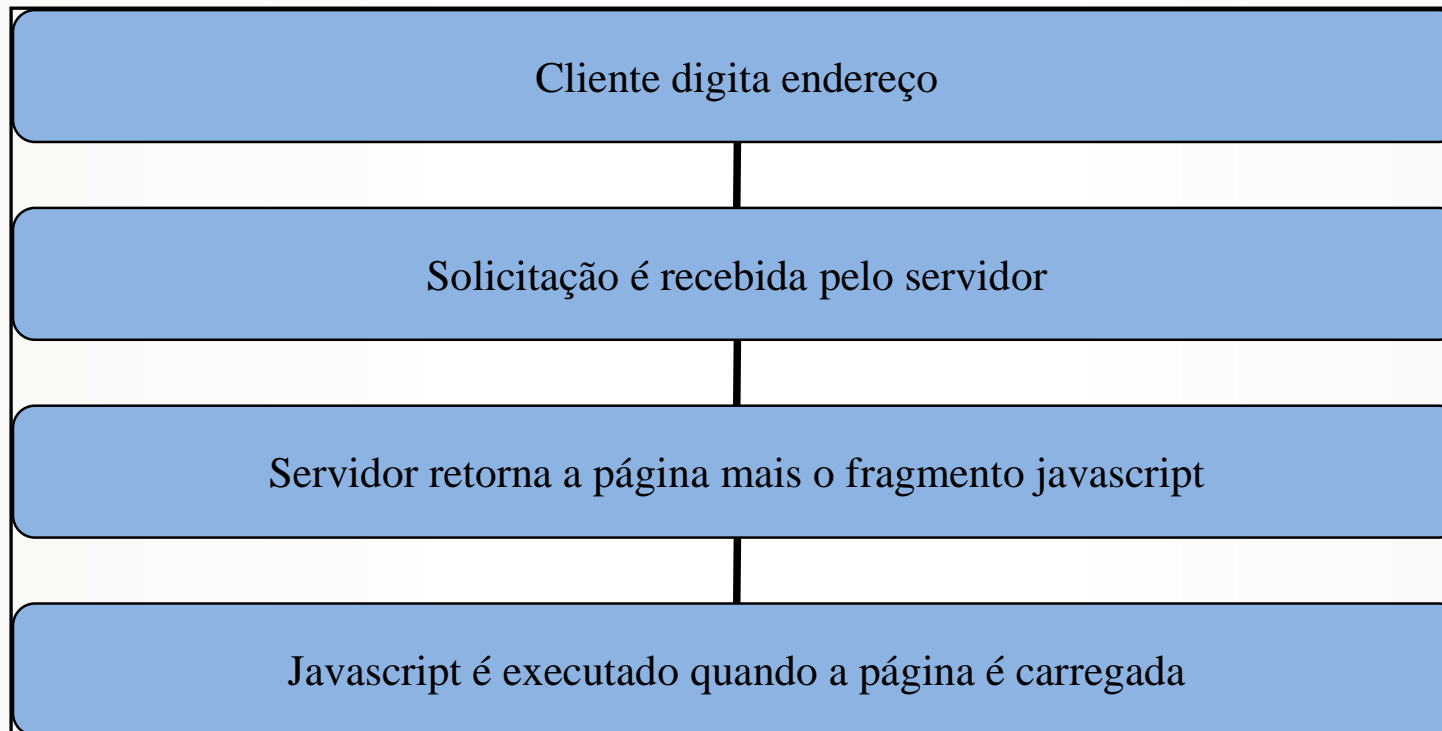
```

```



## 2.1.1 Fluxo de cliques

- tags JavaScript





## 2.1.1 Fluxo de cliques

- sniffers de pacotes
  - intercepta o tráfego

## 2.1.2 Metodologias Web Analytics

### Site Centric

- medição da máquina do usuário através de um senso
- capacidade de medir todos os usuários de seu web site

### User Centric

- medição do usuário de internet através de um painel de colaboradores (pesquisa)

Métricas básicas que podemos analisar nos sites e na mídia online





## 2.1.3 Behavior Targeting

- significa automaticamente que seus dados ficarão cada vez mais relevantes quando alguns fatores que os “sujam”, tornarem-se uma parte menor do todo, conforme Kaushik (2009, p. 12).
- é preciso mapear preferências para valores numéricos

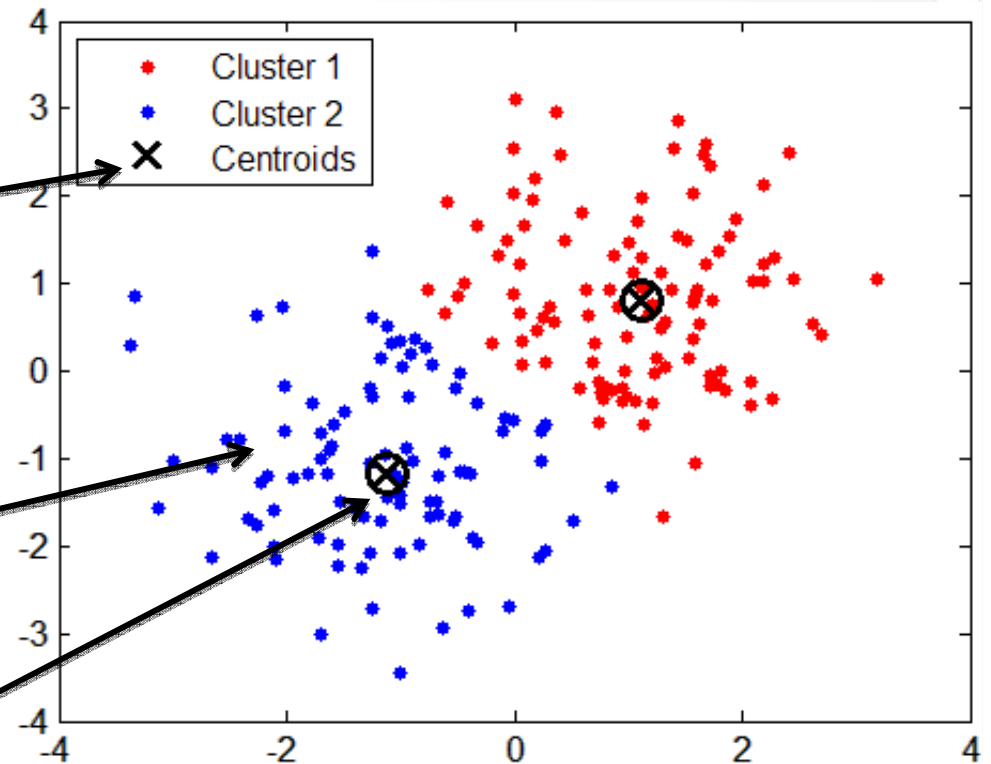


## 2.2 DATA MINING

- “Data Mining é o processo de descobrir padrões em dados.” (WITTEN, FRANK, 1999, p. 7, tradução nossa).
- A ideia do algoritmo k-means (também chamado de k-médias) é fornecer uma classificação de informações de acordo com os próprios dados” (PICHILIANI, 2008).

## 2.2 DATA MINING – KMEANS

- permite dividir um determinado conjunto de dados em grupos na ordem de  $K$
- Seleciona-se  $K$  centros de clusters iniciais e em seguida iterativamente o algoritmo refina-se da seguinte maneira:
  - Cada instância  $d_i$  é atribuída a seu centro de clusters mais próximos, os centroids;
  - Cada centro de cluster  $C_j$  é atualizado para ser o meio de suas instâncias constituintes.





## 2.2.1 WEKA

- Weka é uma coleção de algoritmos de aprendizagem para tarefas de mineração de dados (WITTEN, FRANK, 2009, p. 7, tradução nossa).
- ARFF:
  - @relation
  - @attribute + nome + tipo (numérico ou nominal)
  - @data seguida pelos dados que devem ser separados por vírgula e correspondentes aos atributos





## 2.2.1 WEKA

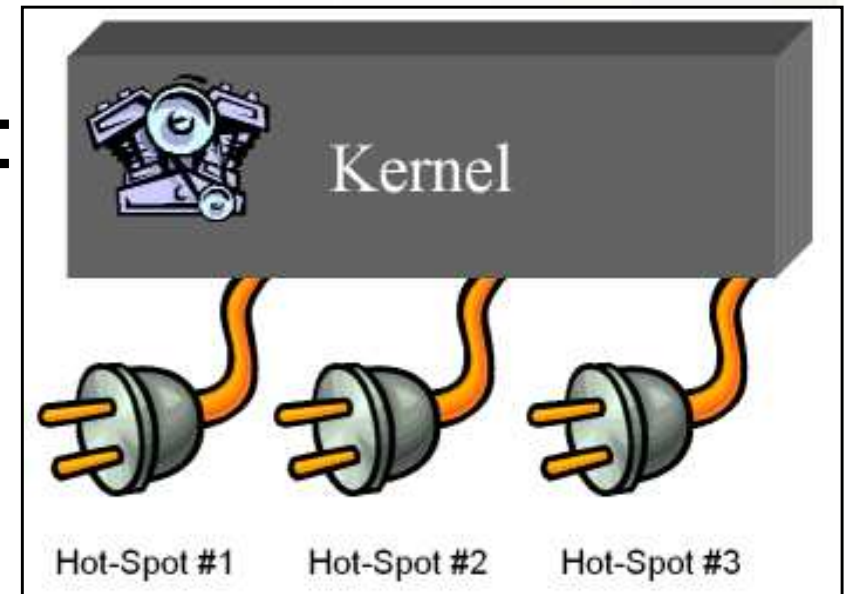
```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%   (a) Creator: R.A. Fisher
%   (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%   (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class       {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

## 2.3 DESENVOLVIMENTO DE FRAMEWORKS

- Segundo Pressman (2006, p. 204), framework – ou arcabouço – é um esqueleto com uma coleção de pontos conectáveis.
- Partes fixas e variáveis:
  - Frozen spot
  - Hot Spot





## 2.4 TRABALHOS CORRELATOS

- 2.4.1 LI, T. et al. A markov chain model for integrating behavioral targeting into contextual advertising
- 2.4.2 WU, X. et al. Probabilistic Latent Semantic User Segmentation for Behavioral Targeted Advertising
- 2.4.3 CHEN, Y. PAVLOV, D. CANNY, J. F. Large-Scale Behavioral Targeting



## 2.4.1 LI, T. et al

- Modelo de Markov para aplicação de Behavior Targeting em estratégias de publicidade
- Entrada de dados:
  - anúncios candidatos
  - o conjunto de páginas visitadas pelos usuários que clicaram nos anúncios candidatos
  - os usuários que navegaram pelas páginas que contém anúncios candidatos



## 2.4.2 Outros exemplos

- WU, X. et al. Probabilistic Latent Semantic User Segmentation for Behavioral Targeted Advertising
  - os termos digitados nas consultas dos usuários são utilizados para sua segmentação.
- CHEN, Y. PAVLOV, D. CANNY, J. F. Large-Scale Behavioral Targeting
  - segmentar comportamentos do usuário web em larga escala usando para isso grids computacionais



# 3 DESENVOLVIMENTO

- 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO
- 3.2 ESPECIFICAÇÃO
- 3.3 IMPLEMENTAÇÃO
- 3.4 ESTUDO DE CASO
- 3.5 RESULTADOS E DISCUSSÃO



## 3.1 REQUISITOS

- a) o framework deverá permitir capturar por usuário as páginas visitadas e a seqüência de visitas (RF);
- b) o framework deverá permitir fornecer como saídas que usuários pertencem aos grupos segmentados (RF);
- c) o framework deverá permitir coletar dados estatísticos utilizando um funil de navegação com Hub-and-Spoke (RNF);
- d) necessita ser desenvolvido baseado em filtros JEE (RNF);
- e) necessita ser compatível com MySQL para persistência dos dados (RNF);
- f) deve utilizar a biblioteca Weka para mineração de dados (RNF);
- g) deve ser compatível com Apache Tomcat (RNF);

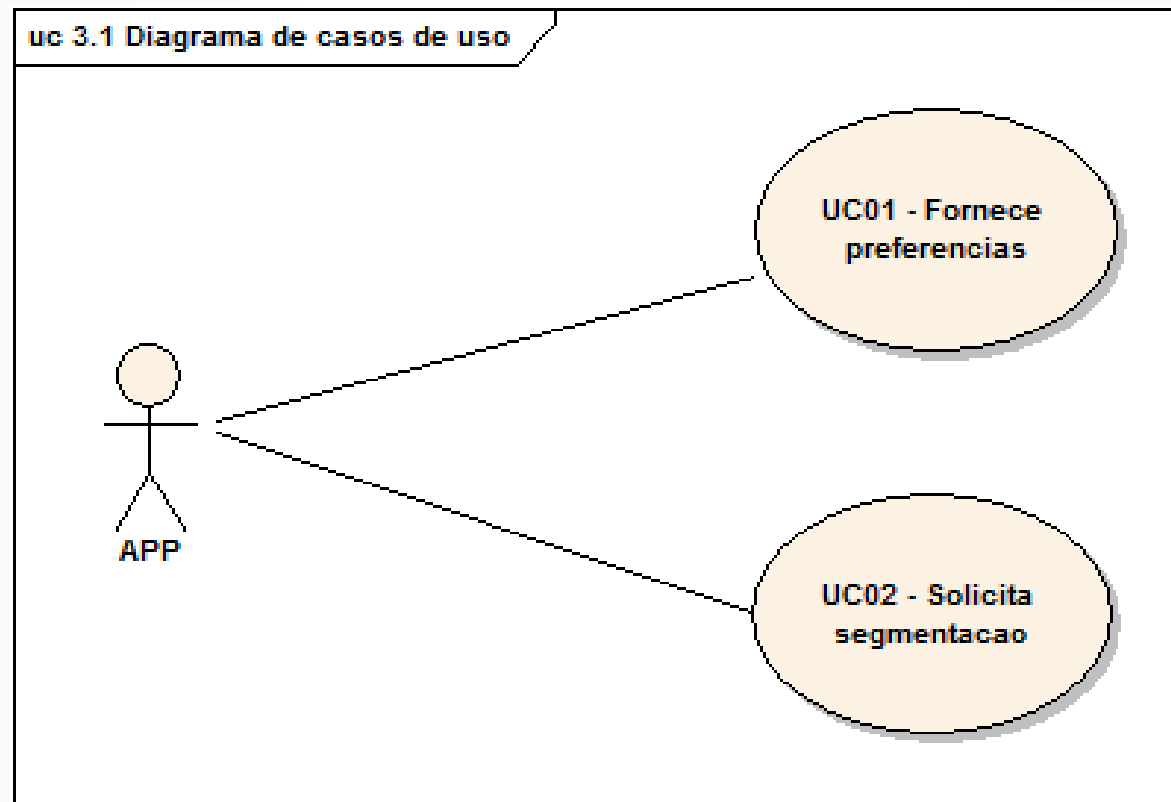


## 3.2 ESPECIFICAÇÃO

- 3.2.1 Diagrama de casos de uso
- 3.2.2 Diagrama de atividades
- 3.2.3 Diagrama de classes
- 3.2.4 Diagrama de sequência
- 3.2.5 Arquitetura do software
- 3.2.6 Modelo Relacional

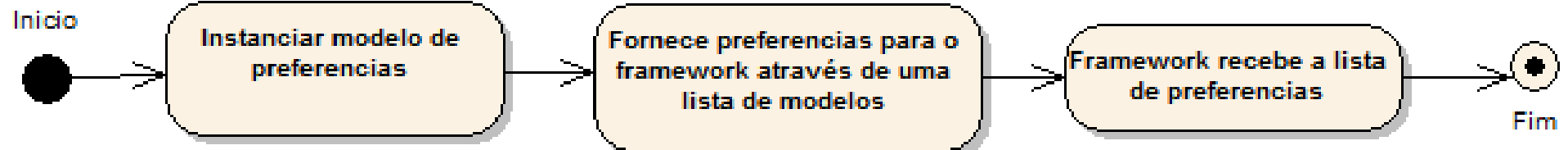


## 3.2.1 Diagrama de casos de uso

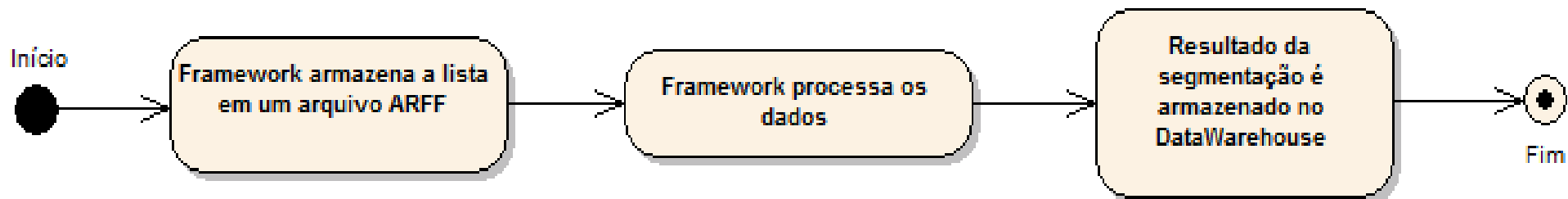


## 3.2.2 Diagrama de atividades

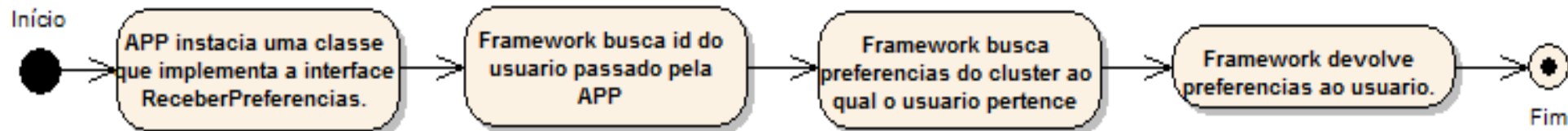
act AT01 - Fornecer preferencias



act AT02 - Segmentar preferencias

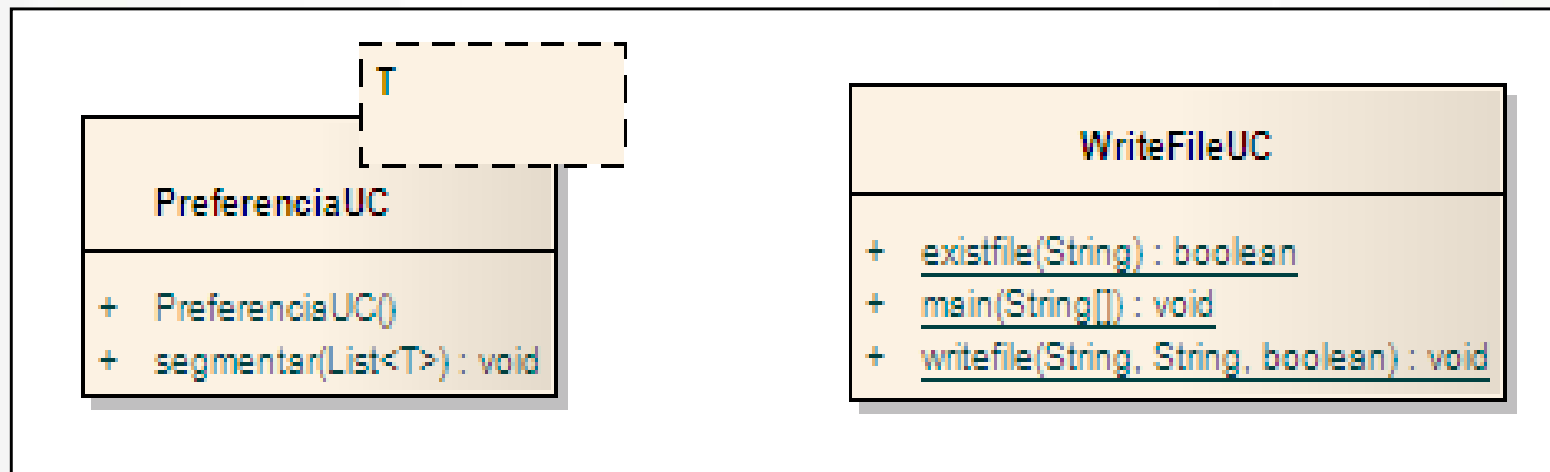


act AT03 - Solicitar segmentacao

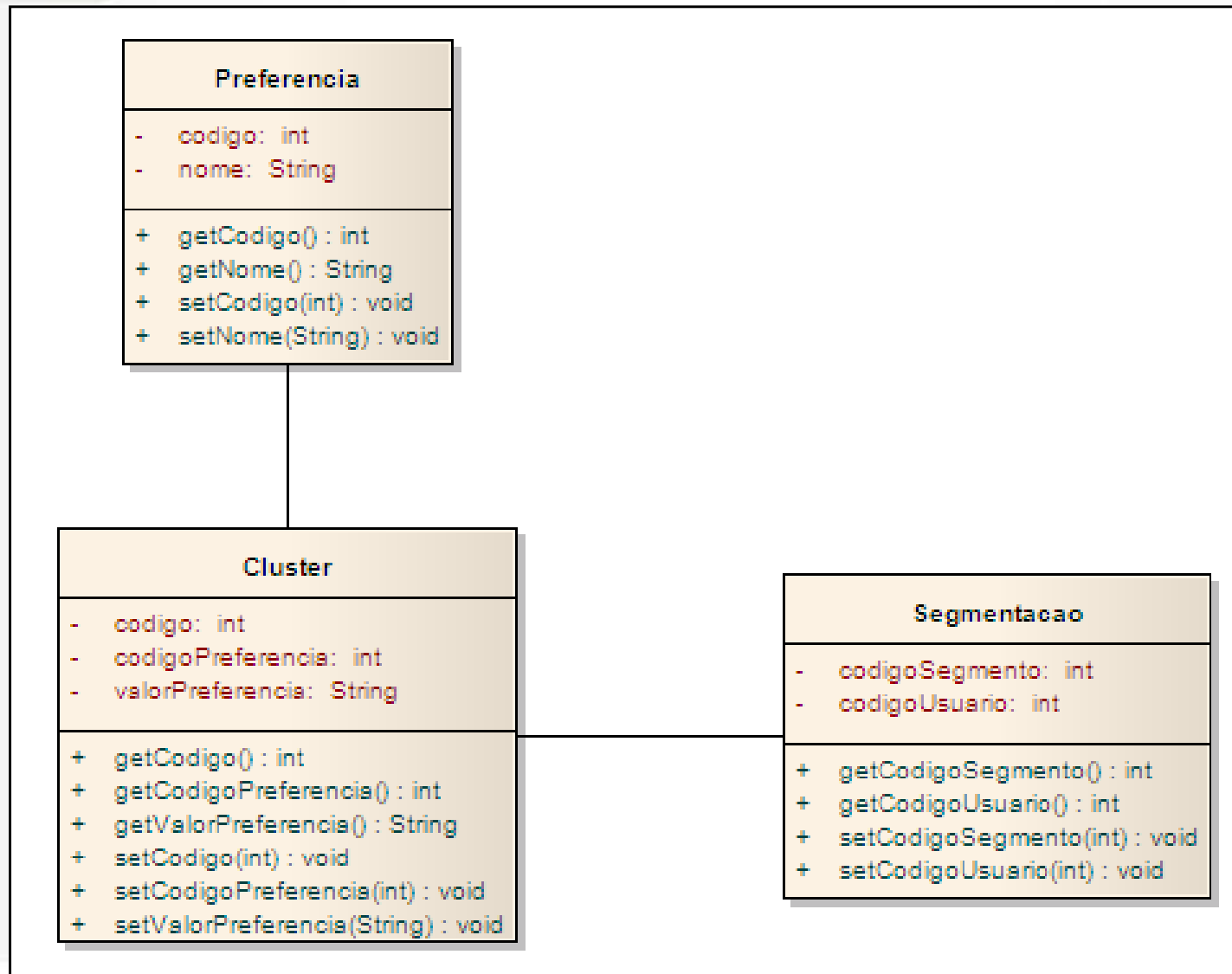




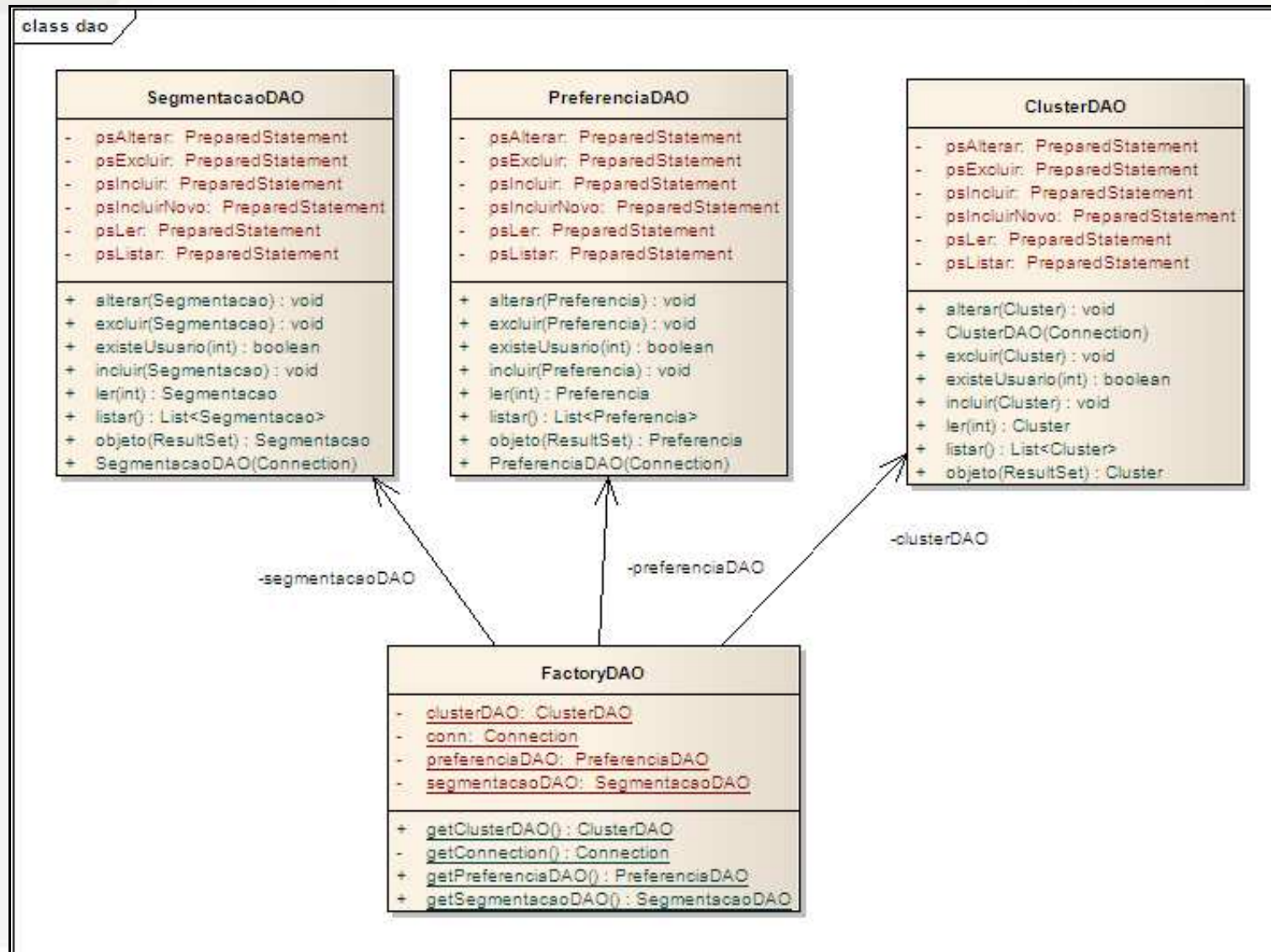
## 3.2.3 Diagrama de classes - Controle



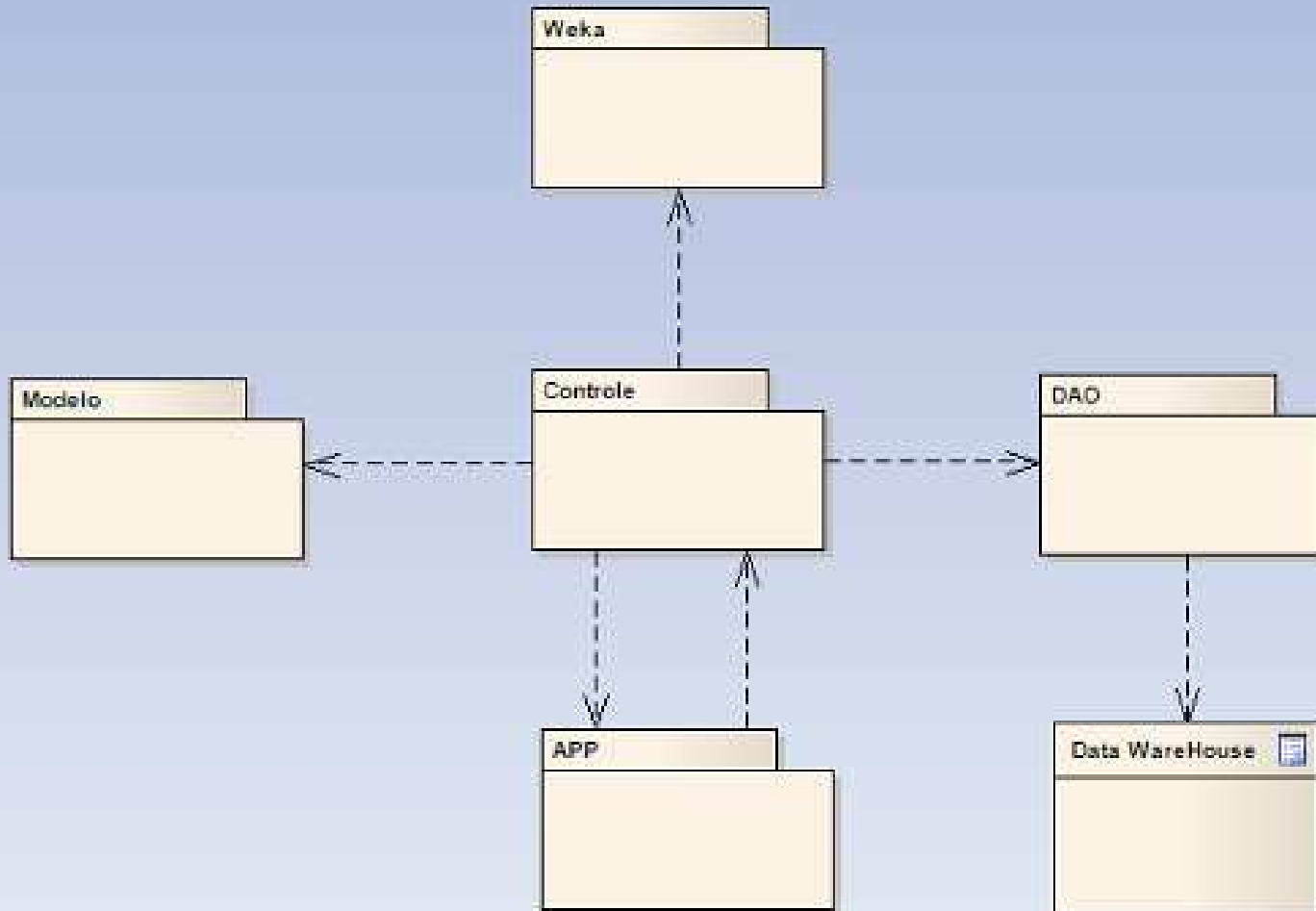
## 3.2.3 Diagrama de classes - Modelo



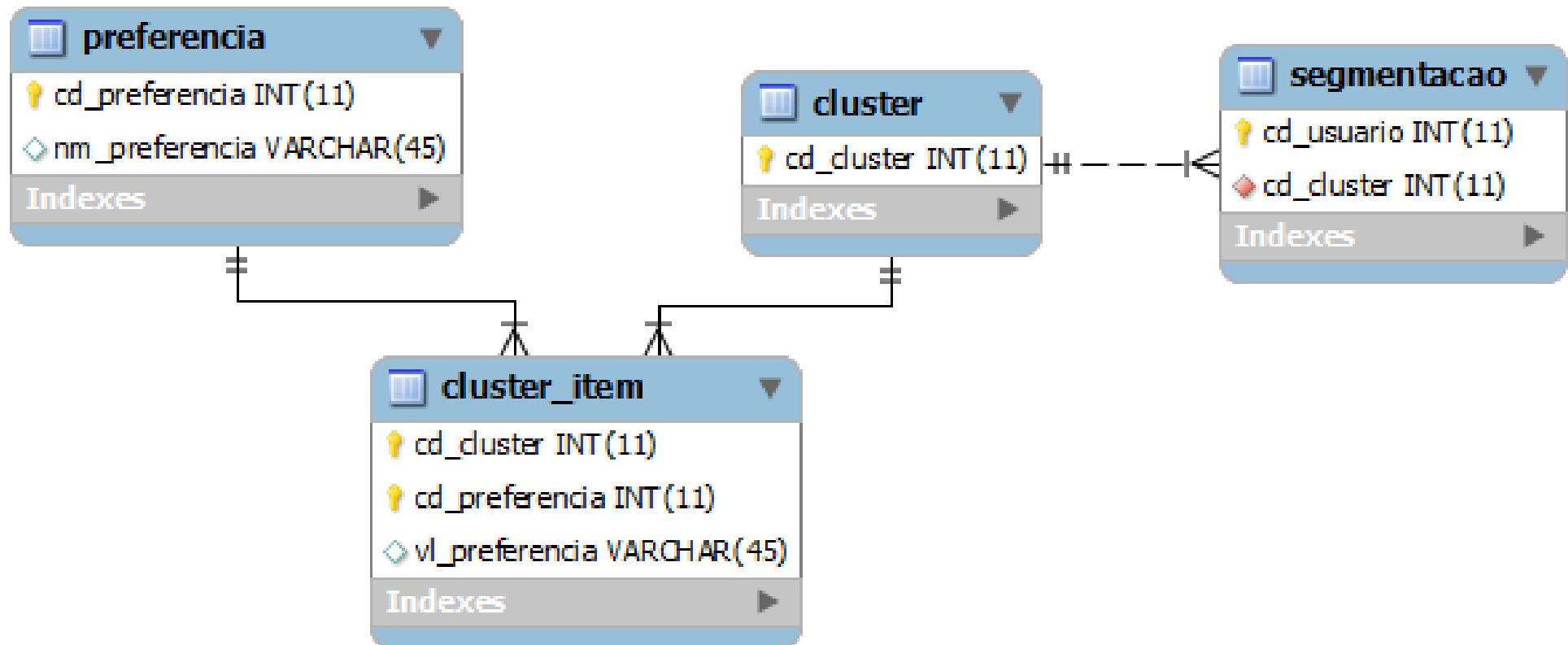
## 3.2.3 Diagrama de classes - DAO



## 3.2.4 Arquitetura do software – Framework



## 3.2.5 Modelo Relacional





## 3.3 IMPLEMENTAÇÃO

- 3.3.1 Tecnologias e padrões utilizados
- 3.3.2 Código fonte





## 3.3.1 Tecnologias e padrões utilizados

- a) linguagem Java;
- b) IDE eclipse;
- c) biblioteca WEKA;
- d) estilo arquitetural em camadas;
- e) reflection;
- f) padrão de projeto DAO;
- g) padrão de projeto Factory;
- h) padrão de projeto Singleton;
- i) banco de dados MySQL.



## 3.3.2 Código fonte

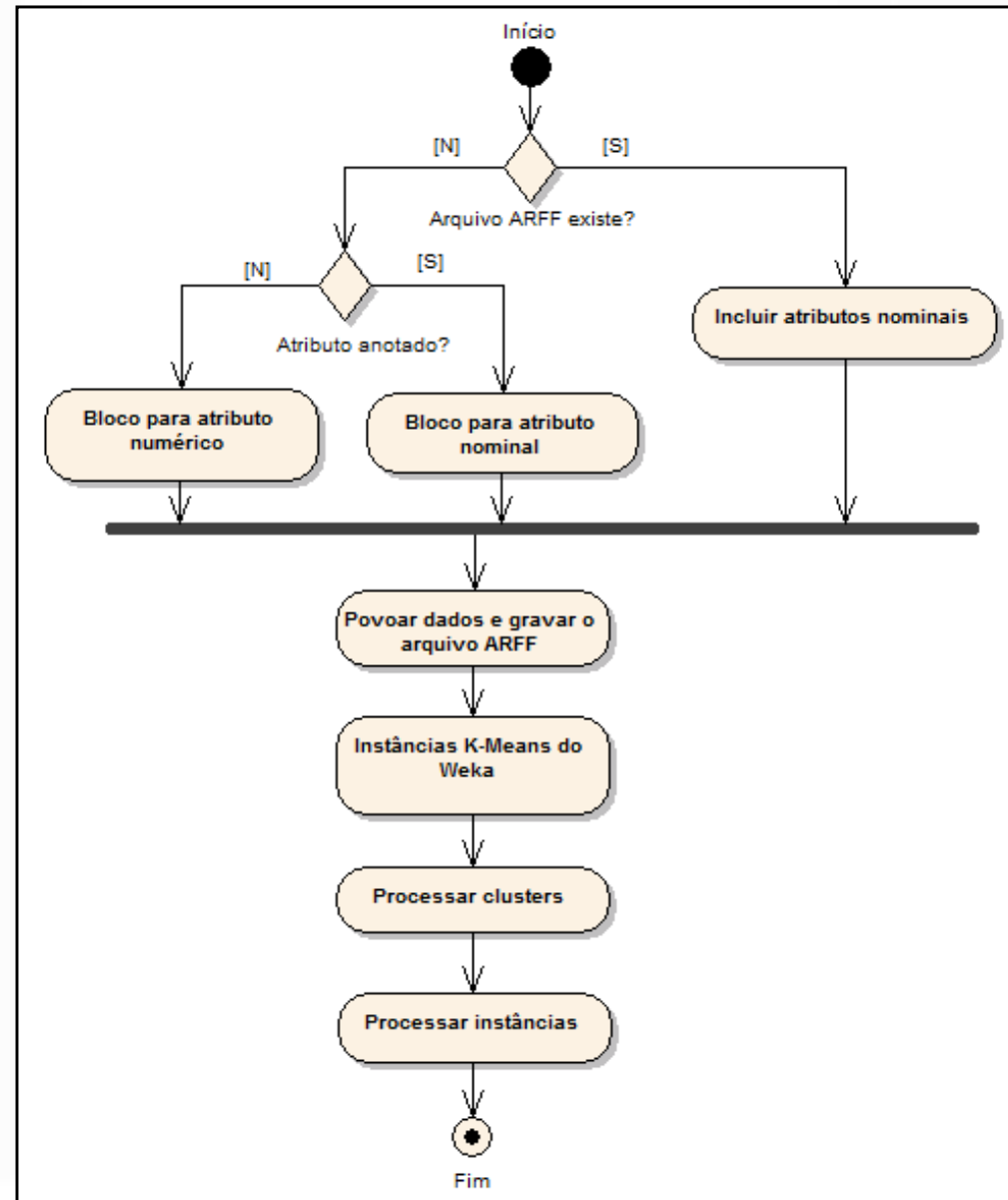
- Dos métodos utilizados os mais relevantes para o trabalho estão presentes na classe PreferenciaUC e são nomeados como:
  - a) segmentar;
  - b) getSegmentacao.



## 3.3.2 Código fonte – Atributos anotados

```
1    public class Preferencia {
2        private int codigo;
3        @AtributoDiscreto
4        private String marca;
5        @AtributoDiscreto
6        private String nome;
7        @AtributoDiscreto
8        private String descricao;
9        @AtributoDiscreto
10       private String tipo;
11       @AtributoDiscreto
12       private String setor;
13       private int validade;
14       private int barras;
15       private int promocao;
16       private Double valor;
```

## 3.3.2 Código fonte – segmentar





## 3.3.2 Código fonte – getSegmentacao

```
1 public List<Preferencia> getSegmentacao(int usuario) throws Exception {
2     SegmentacaoDAO daoSegmentacao = FactoryDAO.getSegmentacaoDAO();
3     Segmentacao segmentacao = daoSegmentacao.ler(usuario);
4     ClusterItemDAO daoClusterItem = FactoryDAO.getClusterItemDAO();
5     List<ClusterItem> clusterItemList =
daoClusterItem.listar(segmentacao.getCodigoCluster());
6     ArrayList<Preferencia> preferencialist = new
ArrayList<Preferencia>();
7     for (ClusterItem clusterItem : clusterItemList) {
8         PreferenciaDAO daoPreferencia =
FactoryDAO.getPreferenciaDAO();
9         Preferencia preferencia = null;
10        preferencia =
daoPreferencia.ler(clusterItem.getCodigoPreferencia());
11        preferencia.setValor(clusterItem.getValorPreferencia());
12        preferencialist.add(preferencia);
13    }
14    return preferencialist;
15 }
```

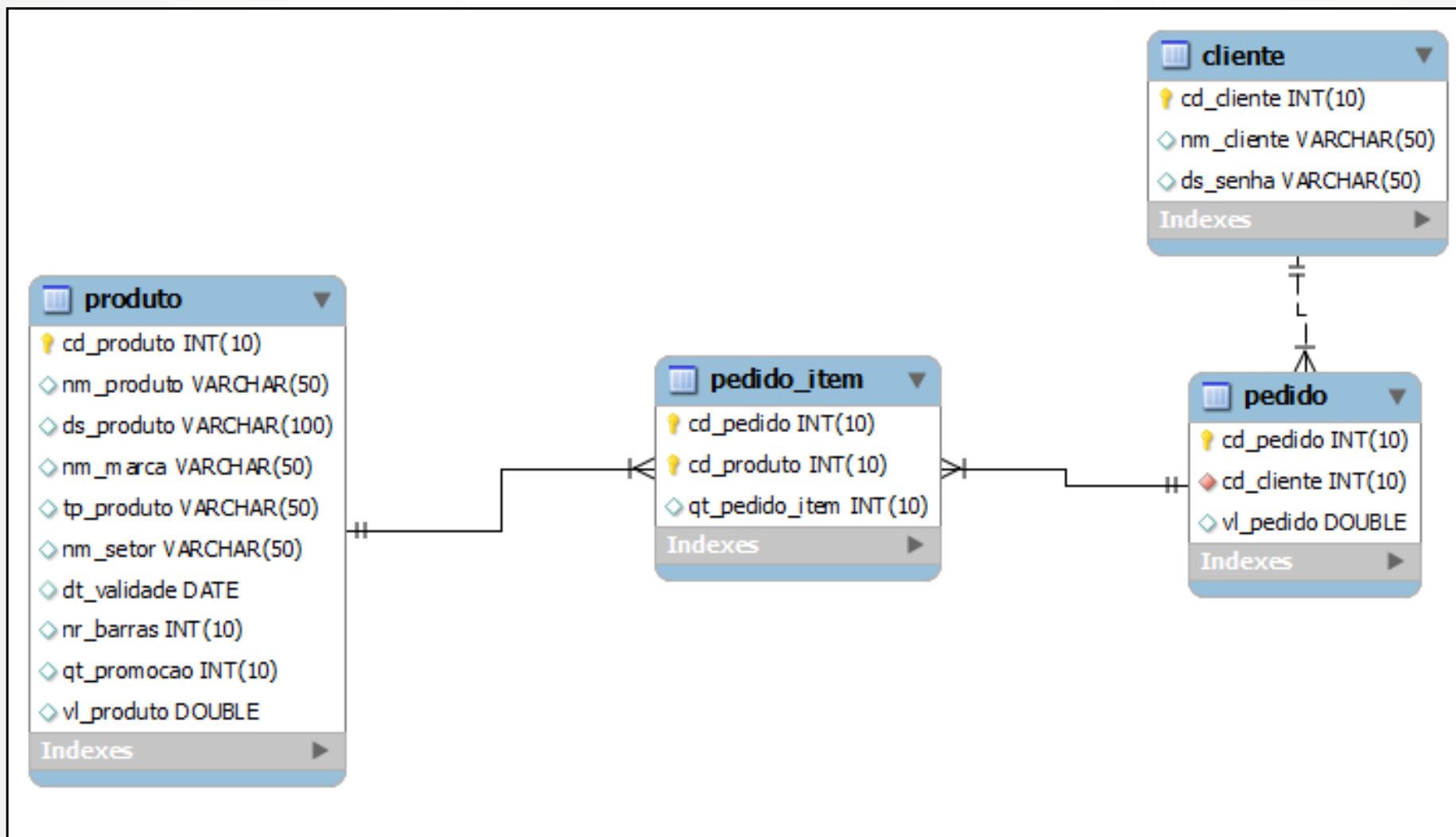


## 3.4 ESTUDO DE CASO – UTILIZAÇÃO DO FRAMEWORK

- UTILIZAÇÃO DO FRAMEWORK
- EXEMPLOS



# 3.4 ESTUDO DE CASO – UTILIZAÇÃO DO FRAMEWORK





## 3.4 ESTUDO DE CASO – UTILIZAÇÃO DO FRAMEWORK

```
1 public void incluir(int codigoPedido, int codigoProduto, int quantidade)
  throws Exception {
2     PedidoItem pedidoItem = new PedidoItem();
3     pedidoItem.setCodigoPedido(codigoPedido);
4     ...
12    Preferencia preferencia = new Preferencia();
13    preferencia.setCodigo(pedido.getCodigoCliente());
14    ...
23    List<Preferencia> listaPreferencia = new ArrayList<Preferencia>();
24    listaPreferencia.add(preferencia);
25    PreferenciaUC<Preferencia> preferenciax = new
  PreferenciaUC<Preferencia>();
26    preferenciax.segmentar(listaPreferencia, 5,
  "C:/Users/RUDIMAR/Documents/beta");
27 }
```





## 3.4 ESTUDO DE CASO - EXEMPLO

http://localhost:8080/beta/sistema.do?action=nc

http://localhost:8080/beta/siste

Nome :

Senha :

http://localhost:8080/beta/sistema.do?action=a

http://localhost:8080/beta/sist

produto1

produto2

produto3

produto4

produto5

[Continuar Comprando](#)

@relation BeTa

@attribute codigo numeric

@attribute marca {marca1, marca2, marca3, marca4, marca5, marca6, marca7, marca8, marca9, marca10, marca11, marca12, marca13, marca14, marca15, marca16, marca17, marca18, marca19, marca20, marca21, marca22, marca23, marca24, marca25, marca26, marca27, marca28, marca29, marca30, marca31, marca32, marca33, marca34, marca35, marca36, marca37, marca38, marca39, marca40, marca41, marca42, marca43, marca44, marca45, marca46, marca47, marca48, marca49, marca50}

@attribute nome {produto50, produto49, produto48, produto47, produto46, produto45, produto44, produto43, produto42, produto41, produto40, produto39, produto38, produto37, produto36, produto35, produto34, produto33, produto32, produto31, produto30, produto29, produto28, produto27, produto26, produto25, produto24, produto23, produto22, produto21, produto20, produto19, produto18, produto17, produto16, produto15, produto14, produto13, produto12, produto11, produto10, produto9, produto8, produto7, produto6, produto5, produto4, produto3, produto2, produto1}

@attribute descricao {descricao50, descricao49, descricao48, descricao47, descricao46, descricao45, descricao44, descricao43, descricao42, descricao41, descricao40, descricao39, descricao38, descricao37, descricao36, descricao35, descricao34, descricao33, descricao32, descricao31, descricao30, descricao29, descricao28, descricao27, descricao26, descricao25, descricao24, descricao23, descricao22, descricao21, descricao20, descricao19, descricao18, descricao17, descricao16, descricao15, descricao14, descricao13, descricao12, descricao11, descricao10, descricao9, descricao8, descricao7, descricao6, descricao5, descricao4, descricao3, descricao2, descricao1}

@attribute tipo {tipo50, tipo49, tipo48, tipo47, tipo46, tipo45, tipo44, tipo43, tipo42, tipo41, tipo40, tipo39, tipo38, tipo37, tipo36, tipo35, tipo34, tipo33, tipo32, tipo31, tipo30, tipo29, tipo28, tipo27, tipo26, tipo25, tipo24, tipo23, tipo22, tipo21, tipo20, tipo19, tipo18, tipo17, tipo16, tipo15, tipo14, tipo13, tipo12, tipo11, tipo10, tipo9, tipo8, tipo7, tipo6, tipo5, tipo4, tipo3, tipo2, tipo1}

@attribute setor {setor1, setor2, setor3, setor4, setor5, setor6, setor7, setor8, setor9, setor10, setor11, setor12, setor13, setor14, setor15, setor16, setor17, setor18, setor19, setor20, setor21, setor22, setor23, setor24, setor25, setor26, setor27, setor28, setor29, setor30, setor31, setor32, setor33, setor34, setor35, setor36, setor37, setor38, setor39, setor40, setor41, setor42, setor43, setor44, setor45, setor46, setor47, setor48, setor49, setor50}

@attribute validade numeric

@attribute barras numeric

@attribute promocao numeric

@attribute valor numeric

@data

39,marca50,produto1,descricao1,tipo1,setor50,1,1,40,10

39,marca49,produto2,descricao2,tipo2,setor49,2,2,30,100



## 3.4 ESTUDO DE CASO - EXEMPLO

cd_cliente	nm_cliente	ds_senha
39	rudi1	rudi1
40	rudi2	rudi2
41	rudi3	rudi3
42	rudi4	rudi4
43	rudi5	rudi5
44	rudi6	rudi6
45	rudi7	rudi7
46	rudi8	rudi8
47	rudi9	rudi9
48	rudi10	rudi10

cd_usuario	cd_cluster
39	5
40	1
41	1
42	4
43	5
44	1
45	3
46	4
47	2
48	3

cd_cluster	cd_preferencia	vl_preferencia
5	1	41.777778
5	2	marca18
5	3	produto33
5	4	descricao33
5	5	tipo33
5	6	setor18
5	7	1
5	8	17
5	9	40
5	10	10

cd_usuario	cd_cluster
39	4
40	3
41	3
42	5
43	1
44	1
45	1
46	5
47	2
48	2



## 3.4 ESTUDO DE CASO – UTILIZAÇÃO DO FRAMEWORK

```
1 public Logon ler(String nome, String senha) throws Exception {  
2     LogonDAO dao = FactoryDAO.getLogonDAO();  
3     Logon logon = dao.ler(nome, senha);  
4     PreferenciaUC<app.modelo.Preferencia> preferenciax = new  
PreferenciaUC<app.modelo.Preferencia>();  
5     List<framework.modelo.Preferencia> preferencialist =  
preferenciax.getSegmentacao(logon.getCodigoUsuario());  
6     ...  
32     ProdutoDAO produtoDAO = FactoryDAO.getProdutoDAO();  
33     List produtoLista = produtoDAO.listarPreferencia(marca, tipo,  
setor, garantia, promocao, valor);  
34     return logon;  
35 }
```

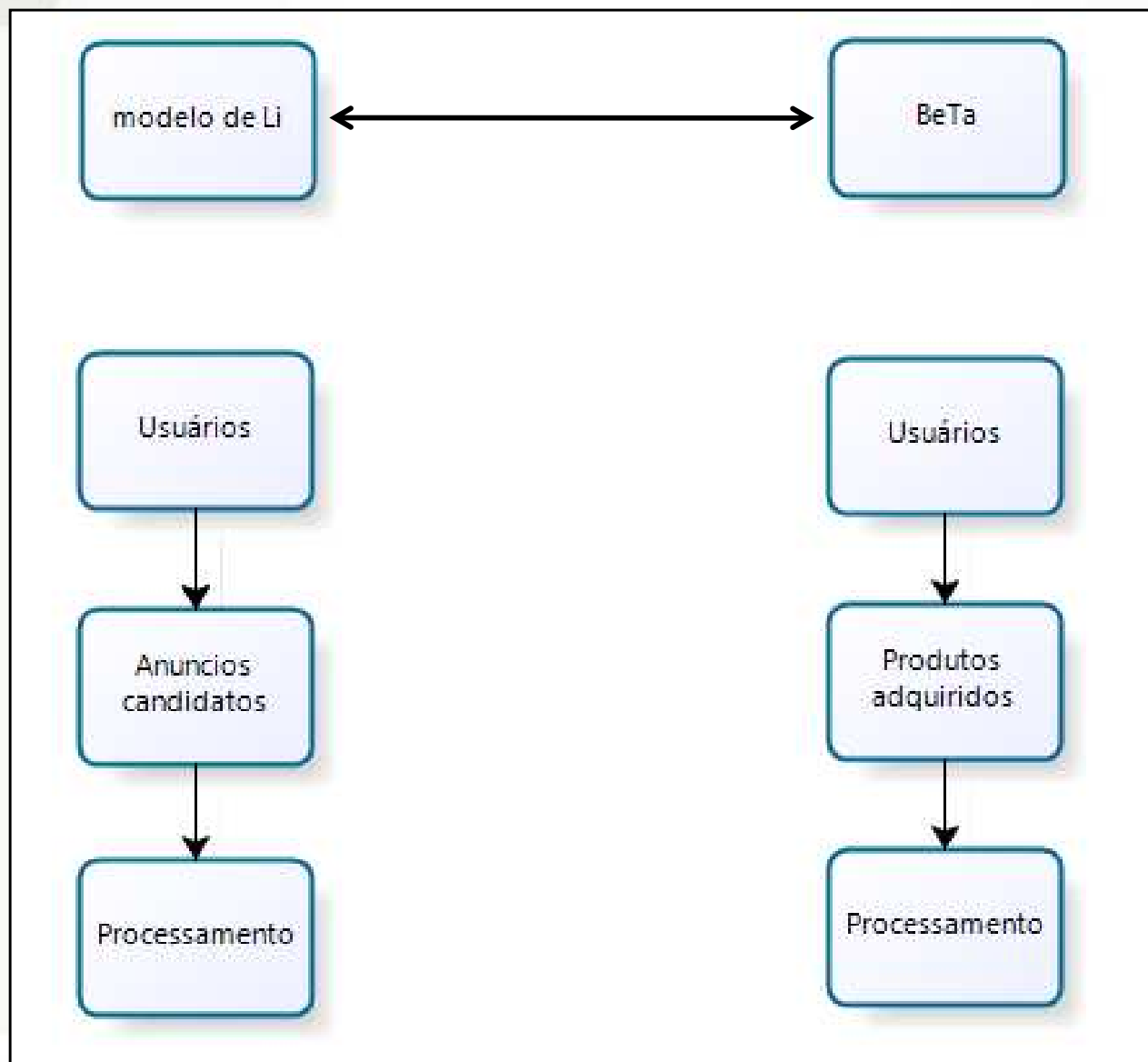


## 3.4 ESTUDO DE CASO – ANALISE

<b>Preferencia</b>	<b>Valor</b>
marca	marca1
tipo	tipo50
setor	setor1
garantia	2
promoção	30
valor	302,5

	nm_marca	tp_produto	nm_setor	dt_validade	qt_promocao	vl_produto
▶	marca50	tipo1	setor50	1	40	10
	marca49	tipo2	setor49	2	30	100
	marca48	tipo3	setor48	3	20	1000
	marca47	tipo4	setor47	4	10	10000
	marca46	tipo5	setor46	1	40	10
	marca5	tipo46	setor5	2	30	100
	marca4	tipo47	setor4	3	20	1000
	marca3	tipo48	setor3	4	10	10000
	marca2	tipo49	setor2	1	40	10
	marca1	tipo50	setor1	2	30	100

# 3.5 RESULTADOS E DISCUSSÃO





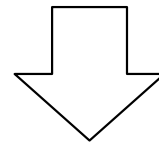
## 3.5 RESULTADOS E DISCUSSÃO

Prós:

- Portabilidade
- Manutenibilidade – uso de padrões
- Objetivos iniciais alcançados

Contras:

- Escalabilidade baixa
- Performance gradativa em relação a escalabilidade



REGULAR



## 4 CONCLUSÕES

- É possível adaptar-se ao usuário e aproximar-se de suas preferências
- O método segmentar é capaz de fazer o processamento necessário as segmentações
- O método getSegmentacao é capaz de devolver o resultado deste processamento



## 4.1 EXTENSÕES

- Baixa escalabilidade
  - Troca do arquivo ARFF por acesso direto a banco de dados
- K-means com resultado diferente a cada execução
  - Busca de outro algoritmo de IA ainda não pesquisado no contexto deste trabalho





# REFERÊNCIAS BIBLIOGRÁFICAS

- CHEN, Y. PAVLOV, D. CANNY, J. F. Large-Scale Behavioral Targeting. **Association for Computing Machinery**, San Jose, 2009. Disponível em: <<http://portal.acm.org/citation.cfm?id=1557019.1557048&coll=portal&dl=GUIDE&CFID=8056571&CFTOKEN=65725324>>. Acesso em: 11 setembro 2009.
- GLOBAL Professor. **Distância Euclidiana** [2010]<[http://professorglobal.cbpf.br/mediawiki/index.php/Dist%C3%A2ncia\\_Euclidiana](http://professorglobal.cbpf.br/mediawiki/index.php/Dist%C3%A2ncia_Euclidiana)>. Acesso em: 25 maio 2010.
- GOMES, Augusto. OLIVEIRA, Kathia, Ana Regina. ROCHA. **Avaliação de Processos de Software Baseada em Medições**. COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação. Disponível em: <<http://www.lbd.dcc.ufmg.br:8080/colecoes/sbes/2001/006.pdf>>. Acesso em: 21 fevereiro 2010.
- GONÇALVES, E. C. Data Mining de regras de associação – Parte 1. **SQL Magazine**, Porto Alegre, 2008. Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=6533>>. Acesso em: 05 sep. 2009.
- JAIN, A. K. and DUBES, R. C. **Algorithms for Clustering Data**. Prentice-Hall, 1988.
- KAUSHIK, Avinash. **Web analytics: uma hora por dia**. 2. ed. Rio de Janeiro : Alta Books, 2009. xxx, 410 p, il.
- LANGER, Sergio. **As métricas da web** < <http://www.slideshare.net/gestaohipermidia/as-mtricas-da-web-1136298>>. Acesso em: 29 fevereiro 2010.



# REFERÊNCIAS BIBLIOGRÁFICAS

- LI, T. et al. A markov chain model for integrating behavioral targeting into contextual advertising. **Association for Computing Machinery**, Beijing, 2009. Disponível em: <<http://portal.acm.org/citation.cfm?id=1592748.1592750&coll=portal&dl=GUIDE&CFID=8056571&CFTOKEN=65725324>>. Acesso em: 11 setembro 2009.
- MARKIEWICZ, Marcus. Eduardo. LUCENA. Carlos J.P. de. . **Association for Computing Machinery**. Disponível em: <<http://www.acm.org/crossroads/espanol/xrds7-4/frameworks.html>>. Acesso em: 20 setembro 2009.
- MATHWORKS The, Disponível em: <<http://www.mathworks.com/access/helpdesk/help/toolbox/stats/kmeans.gif>>. Acesso em: 15 março 2010.
- MELO, C. T. de **Web Beacon**. Disponível em: <[http://imasters.uol.com.br/artigo/13389/desenvolvimento/web\\_beacon/](http://imasters.uol.com.br/artigo/13389/desenvolvimento/web_beacon/)>. Acesso em: 13 maio 2010.
- NUNES, J. L. V. B. **WEB 3.0, o que há de novo?**. Porto Alegre, [2008]. Disponível em: <<http://www.linhadecodigo.com.br/Artigo.aspx?id=1775>>. Acesso em: 25 agosto 2009.
- PICHILIANI, M. Data Mining na prática: algoritmo k-means. **SQL Magazine**, Porto Alegre, 2008. Disponível em: <[http://www.devmedia.com.br/articles/viewcomp.asp?comp=4584&hl=\\*k-means\\*](http://www.devmedia.com.br/articles/viewcomp.asp?comp=4584&hl=*k-means*)>. Acesso em: 05 setembro 2009.
- PRESSMAN, Roger S. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006.
- RIBEIRO, G. **Web Analytics: analisando os números e gerando resultados**. 2. ed. São Paulo: Creative Commons, 2009.
- ROCHA, A. D. Construindo Frameworks em Java. **Java Magazine**, Rio de Janeiro, v. 1, n. 66, p. 76-82, 02. 2009.



# REFERÊNCIAS BIBLIOGRÁFICAS

- ROROHICO, Tari. **Attribute-Relation File Format (ARFF)**. The University of Waikato, New Zealand. 2008. Disponível em: <<http://www.cs.waikato.ac.nz/~ml/weka/arff.html>>. Acesso em: 10 fevereiro 2010.
- SANTOS, Rafael. **Weka na Munheca**. 2005. Disponível em: <<http://www.lac.inpe.br/~rafael.santos/Docs/CAP359/2005/weka.pdf>>. Acesso em: 10 fevereiro 2010.
- SANTOS, Simone M. and NORONHA, Claudio P.. **Padrões espaciais de mortalidade e diferenciais sócio-econômicos na cidade do Rio de Janeiro**. 2001, vol.17, n.5. Disponível em: <<http://www.scielo.org/pdf/csp/v17n5/6319.pdf>> Acesso em: 25 maio 2010.
- SEGARAN, Toby. **Programando a inteligência coletiva: desenvolvendo aplicativos Web 2.0 inteligentes**. Rio de Janeiro: Alta Books, 2008.
- SUN Microsystems. **CORE J2EE PATTERNS: intercepting filter**. [2002]. Disponível em: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/InterceptingFilter.html>>. Acesso em: 14 setembro 2009.
- WITTEN, Ian H. FRANK, Eibe. **Data Mining**. United States: Morgan Kaufmann Publishers, 1999.
- \_\_\_\_\_. **WEKA**. Waikato, [2009]. Disponível em: <<http://www.linhadecodigo.com.br/Artigo.aspx?id=1775>>. Acesso em: 05 setembro. 2009.
- WU, X. et al. Probabilistic Latent Semantic User Segmentation for Behavioral Targeted Advertising. **Association for Computing Machinery**, Beijing, 2009. Disponível em: <<http://portal.acm.org/citation.cfm?id=1592748.1592751&coll=portal&dl=GUIDE&CFID=8056571&CFTOKEN=65725324>>. Acesso em: 11 setembro 2009