CENTRO UNIVERSITÁRIO DA FEI
ANDREY ARAUJO MASIERO

ALGORITMO DE AGRUPAMENTO POR SIMILARIDADE APLICADO A CRIAÇÃO DE PERSONAS

São Bernardo do Campo
2013

ANDREY ARAUJO MASIERO

Algoritmo de Agrupamento por Similaridade aplicado a Criação de Personas

Dissertação de Mestrado apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica, orientado pelo Prof. Dr. Plinio Thomaz Aquino Junior.

São Bernardo do Campo
2013

Masiero, Andrey Araujo
Algoritmo de agrupamento por similaridade aplicado a criação de personas / Andrey Araujo Masiero. São Bernardo do Campo, 2012.
97 f. : il.

Dissertação - Centro Universitário da FEI.
Orientador: Prof. Dr. Plinio Thomaz Aquino Junior.

1. Modelagem de Usuário. 2. Personas. 3. Algoritmo de Agrupamento de Dados. 4. Agrupamento por Similaridade. 5. Q-Sim. I. Aquino Junior, Plinio Thomaz, orient. II. Título.

CDU 681.3.06

Centro Universitário da **FEI**

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA JULGADORA

PGE-10

Programa de Mestrado de Engenharia Elétrica

Aluno: Andrey Araujo Masiero **Matrícula:** 111111-1

Título do Trabalho: Algoritmo de agrupamento por similaridade aplicado a criação de personas.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Plinio Thomaz Aquino Júnior

Data da realização da defesa: 19/02/2013

ORIGINAL ASSINADA

A Banca Julgadora abaixo-assinada atribuiu ao aluno o seguinte:

APROVADO ☒ REPROVADO ☐

São Bernardo do Campo, 19 de Fevereiro de 2013.

MEMBROS DA BANCA JULGADORA

Prof. Dr. Plinio Thomaz Aquino Júnior

Ass.: _____

Prof. Dr. Rodrigo Filev Maia

Ass.: _____

Prof.ª Dr.ª Lucia Vilela Leite Filgueiras

Ass.: _____

VERSÃO FINAL DA DISSERTAÇÃO

ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

A Deus e a minha família que são
o alicerce de minha vida.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer a Deus, que sempre me trouxe sabedoria e luz, mesmo nos momentos difíceis dessa jornada e de tantas outras.

À minha mãe Kathia, que desde o primeiro momento me apoiou e incentivou, mesmo quando tudo parecia impossível.

À minha irmã Andressa, que me suportou quando fiquei exaltado de felicidade ou tristeza perante as dificuldades.

Ao meu orientador Plinio Thomaz Aquino Junior, que me auxilia a direcionar nos caminhos ao longo da jornada acadêmica e pessoal, com seus sábios conselhos e cumplicidade em todos os momentos.

Ao professor Flavio Tonidandel, que ajudou a tornar esse trabalho possível, com seus conselhos e ensinamentos proporcionou um parceria majestosa.

Aos meus avós, Hélio e Rachel, que mesmo não presentes em carne, continuam iluminando minha vida aonde quer que eu vá.

Aos professores do mestrado, que compartilharam ao longo desse período seus conhecimentos e amizade, ajudando na evolução desse trabalho.

Aos meus amigos, que sem esse laço seria impossível avançar mais um passo neste caminho. Os momentos de descontração, de discussão, almoços e cafés foram e são de extrema importância para nos ajudar a andar no caminho chamado vida.

E por fim a todos que de certa maneira contribuíram para mais essa conquista.

“I don’t know anything, but I do know that everything is interesting if you go into it deeply enough.”
Richard Feynman, 1965

RESUMO

A ascensão da tecnologia tem se mostrado presente ao longo dos últimos anos, fazendo com que a quantidade de dispositivos interconectados e seus tipos aumentem significativamente e, como consequência a diversidade dos usuários. Dessa maneira, os projetistas de interface se deparam com o problema de identificar quais são os tipos de usuários do produto em desenvolvimento de tal forma, que suas necessidades e preferências sejam atendidas. Para orientar o desenvolvimento do produto, uma técnica empregada por projetistas é a utilização de perfis que podem carregar consigo as informações dos usuários necessárias ao projeto. Durante a coleta das características do usuário, principalmente do comportamento, a quantidade de informações armazenadas é alta e difícil de interpretar sem ajuda computacional. Para auxiliar na análise e interpretação das informações coletadas, utiliza-se da técnica de clustering para que seja possível o agrupamento dos perfis diminuindo o volume dos dados para análise, mas não diminuindo a qualidade das informações. Todavia, os algoritmos de *clustering* existentes possuem algumas deficiências para o trabalho com perfis de usuários, principalmente. Pensando nisso, essa dissertação propõe um algoritmo que a partir das informações do comportamento do usuário capturadas automaticamente, realiza-se o agrupamento dos perfis com base em um valor de similaridade Q e em seguida apresentam-se os grupos obtidos que apoiam a criação das Personas. Para apoiar o desenvolvimento desta dissertação, uma aplicação prática é realizada no desenvolvimento de um sistema de prontuário eletrônico junto ao projeto Pesquisa e Estatística baseada em Acervo Digital de Prontuário Médico do Paciente em Telemedicina centrada no Usuário (PEAP-PMPT) junto ao Hospital Heliópolis de São Paulo onde o algoritmo de agrupamento por similaridade é aplicado na criação de Personas.

Palavras-chave: Modelagem de Usuário, Personas, Algoritmo de Agrupamento de Dados, Agrupamento por Similaridade, Q-Sim

ABSTRACT

Technology's ascent has proven over the last few years, causing the number of interconnected devices and their types will increase significantly, and as a consequence of the diversity of users. Thus, the interface designers are faced with a problem of identifying which types of users of the product development so that their needs and preferences are met. To guide the product's development a technique used by designers is use of profiles that can carry users' information necessary to the project. During the collection of user's characteristics, mainly the behavior, the amount of stored information is high and hard to interpret without computational aid. To assist in the analysis and interpretation of information collected, we use the clustering technique to make possible to group the profiles by reducing the volume of data to analyze, but not reducing the quality of information. However, existing clustering algorithms have some deficiencies to work with users' profiles, mostly. Thinking about it, this dissertation proposes an algorithm based on information of the user's behavior automatically captured, carried out by the group of profiles based on a similarity value Q and then presents the groups obtained like Personas. To support the development of this thesis, a real application is made during the development of a electronic medical record system in the project Research and Statistics-based Digital Archive of Medical Record of Patient-centered Telemedicine User (PEAP-PMPT) next to the Heliopolis Hospital of São Paulo where the methodology supported by the algorithm will be applied.

Keywords: User Modeling, Personas, Data Clustering Algorithm, Similarity Clustering, Q-Sim

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS

IHC : Interação Humano-Computador

Q-SIM : Algoritmo de Agrupamento por Similaridade com Qualidade (*Quality Similarity Clustering*)

PEAD-PMPT : Pesquisa e Estatística baseada em Acervo Digital de Prontuário Médico do Paciente em Telemedicina centrada no Usuário

FINEP : Financiadora de Estudos e Projetos

SUMÁRIO

1 INTRODUÇÃO

Durante os últimos anos, a área de tecnologia tem disponibilizado um grande número de produtos para o mercado, ampliando os meios de acesso às informações encontradas nos sistemas. O aumento na quantidade de sistemas faz com que haja um crescimento no número de usuários e consequentemente na diversidade do público.

O crescimento da diversidade traz um problema aos projetistas de interface que é como identificar os tipos de usuários, e, ainda atender o aumento dos tipos de usuários durante a execução do projeto. Uma das maneiras para tratar o problema da diversidade é segmentar o público utilizando perfis, que trazem as informações necessárias para projetar sistemas em desenvolvimento e considerar o consumidor do produto.

Em geral, perfis de usuários, possuem informações que caracterizam o usuário, tais como informações etnográficas, de comportamento e/ou habilidade, e auxiliam no conhecimento sobre o usuário para o qual será desenvolvido o sistema (AQUINO JUNIOR; FILGUEIRAS, 2005). Tais dados são importantes para identificar as preferências e necessidades do público alvo do sistema em desenvolvimento. Com o conhecimento dessas informações é possível identificar qual componente ou aparência para o sistema é mais adequada ao usuário proporcionando uma melhor experiência durante o seu uso.

Para capturar as informações que compõem um perfil podem-se utilizar diversos métodos. Os principais são: (I) a utilização de questionários; (II) a realização de entrevista com o usuário; e (III) a utilização de coleta dos dados automatizada através dos *logs* de sistema, que são capazes de capturar as ações do usuário em tempo de uso. Quando se utiliza os dois primeiros métodos citados é possível identificar informações etnográficas e ainda o comportamento declarado do usuário. Utilizando o *log* é possível obter informações sobre o comportamento real do usuário, pois a coleta ocorre durante a utilização do sistema de forma transparente ao usuário.

Após coletar as informações dos perfis é necessário efetuar uma análise para identificar informações similares entre os perfis dos usuários, dessa forma, é possível encontrar um conjunto de perfis que caracterizam os usuários, mas diminui a quantia de dados para análise de componentes a serem utilizados. Uma das formas de utilizar o conhecimento dos conjuntos de perfis de usuários é através da técnica de Personas apresentada por Cooper (1999).

Personas tem como objetivo criar um personagem fictício ou um arquétipo hipotético para representar o perfil de um conjunto de usuários reais, gerando assim um modelo de usuário capaz de ser aplicado a um projeto. Dessa forma, é possível atender a uma diversidade grande de usuários durante um projeto de interface, por exemplo, sem a necessidade de analisar individualmente o comportamento e necessidades de todo o público alvo do sistema (COOPER, 1999; PRUITT; ADLIN, 2005; AQUINO JUNIOR, 2008).

Todavia, as analises realizadas para agrupamentos de perfis e consequentemente criação das personas tem ocorrido através de processos que necessitam do tempo e da experiência dos especialistas. Com o objetivo de facilitar o processo e torna-lo automatizado, diversos métodos são apresentados utilizando a técnica de *clustering* ou agrupamento de dados. O algoritmo mais utilizado para realizar a tarefa de agrupamento dos perfis chama-se k-*means*, de acordo com o estudo realizado previamente representado em Masiero et al. (2011). Em Masiero et al. (2011) é apresentado uma metodologia para criação de personas aonde o k-*means* é utilizado para apoiar essa criação.

A utilização do algoritmo k-*means* para agrupar os perfis, apesar do alto emprego do algoritmo, apresenta um problema. Para executar o k-*means* é necessário informar o número de grupos que o especialista deseja obter a partir do dados coletados. Fornecer a informação da quantidade de grupos existentes em uma base de dados é uma tarefa difícil, pois é uma informação que geralmente não está disponível e é difícil de ser visualizadas em bases de dados com um grande número de informações. Para identificar o melhor número de grupos é necessário realizar algumas comparações dos resultados e definir qual é o melhor agrupamento realizado (MASIERO et al., 2011).

Essa situação gera um segundo problema que é a aleatoriedade dos resultados apresentados pelo k-*means*, devido a isso todos os resultados analisados devem ser armazenados de alguma maneira para que possa ser reutilizado, pois esse resultado pode demorar muito tempo para aparecer novamente (MASIERO et al., 2011).

Com o problema da estimativa de grupos apresentada pelo k-*means*, essa dissertação define um novo algoritmo de agrupamento que utiliza como parâmetro o valor da similaridade entre os perfis para identificar os grupos dentro da base de dados coletada. O novo algoritmo de agrupamento por similaridade apresentado nessa dissertação é chamado de Q-SIM.

Essa dissertação tem como aplicação prática o projeto PEAP-PMPT FINEP 01.10.0765.00 junto ao Hospital Heliópolis, onde serão informatizados os prontuários do setor de cirurgia de cabeça e pescoço. Esse projeto tem como objetivo, além da construção de um sistema de prontuário eletrônico, facilitar a consulta aos prontuários e consequentemente à pesquisa na área médica através de um sistema de gestão de documentos centrado no usuário. O foco direcionado a área médica foi dado por tratar-se de uma área com deficiência em sistemas centrados ao usuário, de acordo com o levantamento inicial do projeto.

Um dos motivos para o problema apresentado no projeto é o baixo conhecimento do projetista de interface sobre o conhecimento do usuário médico em sistemas computacionais. Dessa maneira, um componente que coletará as informações do usuário médico será incluído no sistema para que esse conhecimento possa ser gerado. Coletada as informações dos médicos serão obtidas as personas médicas através do algoritmo Q-SIM, apoiando a proposta da dissertação com a análise dos resultados produzidos.

1.1 Motivação

A necessidade de informar um número de grupos de maneira empírica, como é a proposta do k-*means*, torna o processo de criação de personas muito custoso, pois não é sempre que o número de personas é conhecido no projeto. Outro problema em utilizar o algoritmo k-*means* é a análise que deve ser realizada para encontrar a quantidade de grupos existentes na base de perfis.

Esse problema torna difícil a automatização do processo de criação e acompanhamento da evolução das Personas, pois deve-se realizar sempre uma análise dos grupos para identificar se o número de grupos está correto. Isso ocorre mesmo utilizando a coleta de informações sobre o usuário através de *log* do sistema.

1.2 Justificativa

Durante os estudos dos trabalhos que utilizam algoritmos de agrupamento, principalmente o k-*means*, para criação de modelos de usuários notou-se que todos possuem alguma limitação que dificulta a automatização do processo (análise realizada no desenvolvimento deste trabalho). Dessa forma, foi necessária a criação de um novo algoritmo para identificar o grupo de perfis de usuários similares sem a necessidade de analisar os grupos hierárquicos que necessitam da experiência do especialista para definir a melhor junção dos elementos, e nem informar a quantidade de grupos desejada.

Assim, torna-se possível o acompanhamento da evolução dos grupos de usuários e ainda utilizar essas informações para determinar padrões de interface, possibilitando a construções de sistemas com interface adaptativa baseada nas informações de comportamento, habilidade e experiência do usuário.

1.3 Objetivos

Nessa seção são apresentados o objetivo principal e os secundários defendidos por essa dissertação.

1.3.1 Objetivo Principal

Como objetivo principal, essa dissertação define um novo algoritmo de agrupamento por similaridade que identifica a quantidade de grupos existentes na base de dados de perfis de

usuários de maneira automatizada, possibilitando a variação do valor de similaridade entre os perfis dentro dos grupos.

1.3.2 Objetivos Secundários

Os objetivos secundários almejados nessa dissertação são: (I) a aplicação do algoritmo no projeto FINEP; e (II) apresentar uma proposta de metodologia apoiada por um algoritmo que tornará possível a automatização do processo de criação de personas.

1.4 Metodologia

A pesquisa desenvolvida neste trabalho mantém como base os problemas apresentados ao longo da introdução dessa dissertação buscando sempre um algoritmo que crie melhor os grupos de perfis de usuários. A fundamentação do trabalho foi realizada em pesquisas de cada uma das áreas abrangentes, personas e *clustering*, onde identificou-se a possibilidade de automatizar o processo de criação das personas de tal forma, que não houvesse informações ruidosas passadas pelo usuário. Ao analisar os pontos estudados, identificou-se a necessidade de construir um algoritmo que identifique o número de grupos de maneira automática para automatizar o processo de criação de personas baseadas nas informações do usuário.

Com o objetivo definido, realizou-se um estudo referente aos algoritmos de *clustering* que apresentassem um melhor resultado na geração de grupos. Além disso, observaram-se os algoritmos que apresentaram grupos com o maior valor de similaridade entre os seus elementos. Os testes foram realizados baseados em algumas bases de dados consideradas extremas, procurando atender o maior número de casos considerados como problemas para a técnica *clustering*.

Definidos os grupos dos perfis com o auxílio da técnica de *clustering*, algumas medidas de dispersão dos dados são aplicadas dentre os grupos auxiliando na análise das informações dos usuários para a criação das personas. Todas as informações dos perfis utilizadas nesse trabalho tem como foco o comportamento no uso da interface do usuário real, ou seja, aquele coletado através do *log* do sistema.

Finalizadas as etapas acima descritas, o trabalho será incluído como um componente do projeto PEAD-PMPT para realização do caso de estudo e validação do método apresentado nessa dissertação.

1.5 Estrutura do Trabalho

Esta dissertação é composta por um total de 7 capítulos discriminados a seguir.

A seção 1 apresenta a **introdução** do trabalho conduzindo o leitor para o problema que essa pequisa deve contribuir para a mitigação do mesmo.

A seção 2 apresenta o conceito de modelagem de usuário através de **personas** como uma das maneiras de modelagem de usuário.

A seção 3 apresenta do conceito e os algoritmos de ***clustering*** que serviram como base para formação das personas.

A seção 4 apresenta o algoritmo de agrupamento por similaridade - **Q-SIM** - proposto por esta dissertação.

A seção 5 apresenta a **avaliação** do algoritmo Q-SIM.

A seção 6 apresenta os resultados obtidos com a aplicação da metodologia proposta no projeto PEAP-PMPT.

A seção 7 apresenta as **conclusões** obtidas através dos resultados e os **trabalhos futuros** possíveis como continuação deste trabalho.

2 PERSONAS

Diversas áreas de pesquisa e indústria trabalham com modelos para representar comportamentos, estruturas ou relações complexas com o objetivo de um melhor entendimento ou visualização de um cenário que possa ser melhorado. Sem modelos torna-se complicado o aproveitamento das informações que dados coletados tem a oferecer, pois basicamente é realizado um trabalho em informações cruas, que por muitas vezes pode omitir algumas informações que são importantes para o estudo e análise. Modelos são utilizados a muito tempo por físicos, por exemplo, que criam modelos de átomos ou fenômenos naturais em pesquisa. Estes fenômenos naturais são muito complexos e sem a ajuda dos modelos tornaria a tarefa de obter os resultados esperados difícil ou até mesmo impossível. Utilizar modelos traz grandes vantagens, sendo a principal, a ênfase das características de maior relevância e a remoção do foco das características de menor relevância (COOPER; REIMANN; CRONIN, 2007).

Ao projetar uma interface para um determinado sistema, precisa-se entender como será a interação do usuário, quais são suas expectativas para com aquele sistema e quais são suas experiências anteriores. Para entender um pouco melhor as questões, durante a fase de levantamento de requisito ou pesquisa em um projeto ocorrem diversas entrevistas com os usuários para coletar as informações. Posteriormente, são identificadas quais as expectativas de uso, as experiências que o usuário possui em outros sistemas. Esse processo de identificação dos usuários é complexo, pois inclui muitas variáveis. Assim, a utilização de modelos de usuários auxilia no mapeamento das características do usuário, colaborando no projeto de interface (COOPER, 1999).

Durante o processo de mapeamento dos usuários, percebe-se que cada um possui uma pequena diferença entre suas características, necessidades e expectativas para o sistema. Isso dificulta a concepção de uma interface universal que possa atender aos perfis de usuários que utilizarão o sistema. Visualizar essas diferenças e ao mesmo tempo entender a necessidade de cada um dos usuários é uma tarefa muito complexa, sendo assim a criação do modelo facilita essa tarefa. Este modelo é chamado de *personas* (PRUITT; ADLIN, 2005).

Personas auxiliam o entendimento da diversidade entre comportamentos, necessidades e expectativas dos usuários. Esse modelo, não é uma cópia das características de um perfil em específico. Uma interpretação para essa técnica é defini-la como um personagem fictício que representa as características de um grupo de usuários, conforme Aquino Junior (2008). As personas são compostas por informações que identifiquem as necessidades, comportamentos, experiências, entre outras informações, que possam auxiliar durante a concepção da interface ou auxiliem a comunicação sobre o usuário entre os membros da equipe do projeto.

O modelo persona auxilia justamente na criação do personagem fictício, que na maioria das vezes possui um nome, uma descrição biográfica e até mesmo uma foto. Dessa forma, é possível para o projetista de interface assimilar o personagem e desenvolver o projeto focado

nas características deste. Assim, não há motivo para o projetista criar um modelo próprio do usuário, que na maioria das vezes é tendencioso às características do próprio projetista, gerando assim um esteriótipo do especialista como usuário final do sistema (AQUINO JUNIOR, 2008).

A utilização de uma modelagem de usuário, como personas, não significa que o número de usuários considerados no projeto é o maior, mas sim que um pequeno grupo de usuários com necessidades específicas e parecidas são contemplados na persona (COOPER; REIMANN; CRONIN, 2007). Portanto, o projetista de interface pode preocupar-se apenas com as reações que este persona apresentará ao interagir com o sistema. Assim, ao atender as necessidades dessa persona, o projetista da interface estará considerando um grupo maior de usuários na utilização do sistema facilitando o desenvolvimento e minimizando as adaptações que devem ser consideradas na interface (AQUINO JUNIOR, 2008).

Alguns dos benefícios agregados ao projeto devido ao uso de personas são, por exemplo, a determinação do escopo, o auxílio na comunicação entre os membros envolvidos, consenso da equipe por falarem a mesma língua quanto as características do usuário, obter uma medida para a efetividade do projeto, entre outros. Contudo, quando realizar um trabalho com personas deve-se evitar classifica-las em forma de hierarquia, ou seja, classifica-los de acordo com uma importância social, como cargos em uma empresa. Esse tipo de abordagem pode tirar a efetividade do método. Quando as pessoas são classificadas por um papel, os comportamentos, objetivos, expectativas e satisfações tornam-se diferentes, pois são definidas de acordo com o papel que a pessoa deve exercer (COOPER, 1999).

Para criar personas é necessário um processo para observação do cenário de atuação. Os dados coletados a partir desse processo devem ser analisados cuidadosamente, pois quaisquer interferências nestas informações podem gerar ruídos na representação dos usuários a partir das personas obtidas. Para aumentar a efetividade do processo de observação pode-se efetuar uma coleta de informações diretamente com o usuário, como entrevistas, ou até estudos de análise de segmentação mercadológica, entre outros estudos e análises (COOPER, 1999).

Os métodos para coletar informações apresentados anteriormente são eficientes, contudo, por serem invasivos podem gerar também informações ruidosas, pois o usuário ao responder o questionário durante a entrevista pode sentir-se constrangido e responder de tal forma, que a resposta não corresponde a sua verdadeira personalidade por diversos motivos, como por exemplo, vergonha de uma determinada situação vivida (COOPER; REIMANN; CRONIN, 2007).

Para isso, existem métodos de coletar informações sobre características do usuário de maneira menos invasiva, como por exemplo, *logs* de sistema e ferramentas de *eye-tracking*, principalmente. Esses tipos de técnicas auxiliam a adquirir informações do usuário sem que ele fique preocupado com o tipo de resposta ou com o resultado da atividade que será realizada, assim as informações do usuário são obtidas de maneira direta durante a utilização do sistema (COOPER; REIMANN; CRONIN, 2007).

Cooper, Reimann e Cronin (2007) informam que muitos projetistas buscam a reutilização das personas geradas para um determinado produto em outros diferentes, porém a estratégia

de reutilizar personas em diferentes projetos não extrai o ponto forte dessa modelagem de usuário. Quando se cria as personas, é realizado uma análise do comportamento dos usuários para o cenário do sistema em desenvolvimento. Utilizar as personas de um projeto em outro diferente, por menor que seja essa diferença, existe um novo cenário o que leva o usuário a um comportamento, objetivo ou satisfação diferente do anterior. Dessa forma, para usufruir melhor os benefícios da aplicação de personas no projeto, é necessário criar personas exclusivas para cada projeto. Porém, o autor dessa dissertação acredita que é possível identificar variáveis que compõem as personas que possibilitem a reutilização dessas em outros projetos, de tal forma, que seja possível o aprendizado de novos cenários e a adaptação do comportamento das personas.

Personas é um modelo de usuário bem flexível, pois se pode adaptar os atributos ou variáveis da persona para extrair detalhes do comportamento de acordo com o cenário ao qual se projeta o sistema. Esses atributos devem ser escolhidos para melhor representar um personagem ao qual o projetista de interface ficará confortável em interagir. De acordo com Cooper, Reimann e Cronin (2007) quando essas características não trazem esse tipo de informação que auxilia o profissional de interface a determinar os comportamentos, objetivos ou as expectativas do usuário, por exemplo, é preferível inserir informações de gênero, idade, etnia e até mesmo outros dados geográficos, que caracterizam o usuário mais pelas informações demográficas do que as comportamentais. Isso auxilia na composição da persona e torna mais fácil a comunicação entre a equipe através dela.

A tabela 2.1 apresenta um exemplo de persona, que servirá como ilustração para o contexto apresentado até o presente momento.

Uma persona representa o usuário direto do produto ou o usuário final. Porém, pode-se criar uma persona indireta, que interage com o processo, mas não com o produto, como por exemplo, um paciente dentro de uma clínica médica. O paciente tem suas necessidades e motivações, contudo ele não interage com a interface do sistema, mas as necessidades e motivações desse tipo de persona devem ser consideradas durante o projeto da interface. Entretanto, as necessidades e motivações de uma persona indireta não devem influenciar na interface no que diz respeito a questão de interação entre o sistema e o usuário final (COOPER; REIMANN; CRONIN, 2007).

A definição de personas geralmente tende a ser classificada como sinônimo de perfil de usuário por projetistas. Essa definição não está totalmente errada, contudo ao modelar um usuário como perfil, pode-se incluir nas personas informações demográficas, como quantidade de filhos ou local de nascimento, por exemplo. Esse tipo de variável pode atrapalhar, pois não é utilizada na construção da interface, uma vez que não produz nenhum conhecimento para a melhoria da mesma. Dessa forma, o que determina uma persona são as informações sobre seu comportamento, motivação, necessidades, entre outras informações que contribuam para uma melhor interface e não informações demográficas conforme mencionado acima (PRUITT; ADLIN, 2005; COOPER; REIMANN; CRONIN, 2007).

Tabela 2.1 – Exemplo de Persona

Foto:	
Nome:	Dr. Dráuzio
Descrição:	Médico, aos seus 43 anos, é responsável pelo setor de inovação de um grande hospital. Entusiasta de tecnologia gosta de passar seu tempo criando sistemas para automatizar o processo de suas pesquisas. Procura estudar sobre quais os tipos de tecnologias são mais vantajosas para melhorar o desempenho de seu trabalho. Muito preocupado com a segurança das informações e quem tem acesso a elas. Passa algumas horas desenvolvendo alguns aplicativos simples em ambientes como MS Access e utiliza muito contato via e-mail.
Tempo de Navegação no Sistema:	400 segundos
Tempo de Digitação campos textuais:	140 segundos
Solicitação de ajuda ao sistema:	3 vezes

Na sessão 2.1 são apresentados os trabalhos relacionados ao conceito de personas como modelo de usuário. O objetivo dessa sessão é apresentar as diversas possibilidades de construir personas e seus diversos tipos de objetivos. Na sequência uma breve discussão sobre qual a relação deste conceito com essa dissertação de mestrado.

2.1 Utilização de Personas como modelo de usuário

As pesquisas com o tema de personas têm aumentado durante os últimos anos, e tem demonstrado que personas podem ser inseridas em diversos contextos e integradas às diversas técnicas como demonstra o trabalho de Faily e Flechais (2011) que apresenta uma técnica de personas chamada de *Personas Cases*. As *Personas Cases* são personas cujas características são fundamentadas na origem da fonte de dados empíricos. Esse tipo de persona pode ser rastreado para identificar a origem da mesma em qualquer momento do processo.

O motivo pelo qual Faily e Flechais (2011) realizaram esse trabalho, foi que a utilização de entrevistas e abordagens para coleta de dados etnográficos utilizados na maioria dos casos, não são suficientes para validação dos dados dirigidos à persona. Isso ocorre, porque os padrões

comportamentais obtidos por tais métodos são baseados nos tipos de agrupamentos realizados com as informações adquiridas através das entrevistas sobre um determinado assunto. Esse tipo de abordagem induz agrupamentos comportamentais, que de certa maneira, criam descrições narrativas sobre a persona, muitas vezes não verídicas.

Para solucionar esse problema foi utilizado um método para análise qualitativa dos dados chamado *Grounded Theory*. O modelo de *Grounded Theory* utiliza uma coleção de conceitos temáticos e os relacionamentos possíveis que possam existir entre os conceitos (FAILY; FLECHAIS, 2011).

A criação desse tipo de persona necessita relacionar os principais cenários ou conceitos temáticos e como são associados, pois a persona desenvolvida por esse método percorrerá esses caminhos para alcançar os objetivos. O próximo passo no processo de criação das *Personas Cases* é demonstrar e classificar cada relacionamento entre os modelos, e ainda os *claims* que justificam os relacionamentos, representando assim as características potenciais de cada persona. Por último, deve-se criar uma narrativa para cada seção caracterizada por tipos de variáveis comportamentais. Essa narrativa pode ser apoiada por um diagrama de afinidades demonstrando seu agrupamento (FAILY; FLECHAIS, 2011).

Após a definição deste método, Faily e Flechais (2011) realizaram um estudo de caso para validá-lo e em seguida discutir os resultados obtidos. Esse estudo de caso foi realizado em uma empresa de tratamento de água para auxiliar no processo realizado por cada um dos profissionais da empresa. Assim, os autores demonstraram como construir as *Personas Cases* e ainda valida-las de forma independente, utilizando o caminho do processo gerado por elas através da *Grounded Theory*. A aplicação no estudo de caso teve como resultado 3 personas e desejam expandir esse trabalho com o intuito de evoluir essa técnica.

Kan et al. (2010) apresentam em seu trabalho um método para criação de personas ao qual chamam de *Persona-Driven Product Conception Design* (PDPCD), que tem o objetivo de auxiliar o desenvolvimento de produtos. A ideia por trás do PDPCD é construir personas e protótipos de sistemas analisando as preferências das propriedades dos produtos para os usuários representativos e também demonstrar suas correlações.

Esse método é composto por duas partes, a primeira é chamada de *Persona Data Construction* (PDC), onde está o foco do trabalho apresentado por Kan et al. (2010), e a segunda parte é chamada *Persona-Based Query System*. O PDC é composto por três outras partes: (I) extração das características do usuário; (II) extração dos atributos do produto; e (III) extração da preferência do usuário (KAN et al., 2010).

Para realizar o trabalho, Kan et al. (2010) utiliza uma outra definição para persona, diferente da definição de Cooper, Reimann e Cronin (2007). A representação de persona apresentada através da expressão $Persona(Pot, Prf, Rul, Sam)$, onde:

a) **Pot**: são dados biográficos e demográficos, como por exemplo, idade, gênero, profissão, *hobbie*, entre outros.

b) **Prf**: são as informações do comportamento e preferências do usuário.

c) **Rul**: são padrões ou regras para essa persona.

d) **Sam**: são amostras das preferências dessa persona.

Para a construção das personas, Kan et al. (2010) definiram algumas regras para criar o portifólio do produto, onde este combina valores de utilidade das características do produto e as decisões tomadas sobre o produto, que definem a preferência do usuário. Com base no valor de utilidade do produto é realizada a construção das personas através de agrupamentos e estas são comparadas com os centros dos grupos ao qual ela pertence.

A determinação do valor de utilidade é feita através de uma variável que foi nomeada por Kan et al. (2010) como distância da preferência, que tem como função medir a diferença entre os valores de utilidade da amostra em relação aos atributos das personas. Dessa maneira, o algoritmo apresentado procura agrupar os perfis de acordo com as preferências e assim gerar as personas.

Como caso de estudo para o trabalho de Kan et al. (2010), foi utilizado uma base de estudantes, professores e alguns pais de alunos, para validar este método. Essa base reuniu ao todo 700 amostras. Ao final do processo, Kan et al. (2010) percebeu que o número de amostras demonstrou-se insuficiente para o experimento e não foi possível garantir que as personas geradas eram precisas e representativas. Todavia, Kan et al. (2010) afirmam que caso seja possível aumentar o número de amostra o modelo se torna mais abundante e passível de avaliação. Sendo assim, os autores pretendem continuar essa pesquisa, aumentando a base de dados para garantir a efetividade do método e aprimorá-lo se possível.

O trabalho apresentado por Saez e Domingo (2011) é o início da criação de um novo método de aplicação de personas no mercado, pois apesar das personas serem uma técnica difundida no mercado de trabalho, apenas especialistas conseguem criá-las e aplicá-las a projetos. Assim, Saez e Domingo (2011) nesse estudo procuram: (I) reduzir a tendência causada pela utilização de fotos e como alternativa utilizar silhuetas; (II) combinar o conceito de personas com cenários; (III) promover a compreensão dos usuários mostrando os conceitos principais aplicados; e (IV) criar um artefato reutilizável que possa utilizar essas informações em diferentes projetos. Entretanto, o trabalho apresentado é apenas um relatório de como está a situação dos estudos do grupo, então nenhum resultado, seja parcial ou total, foi apresentado.

Personas também podem ser usadas com o foco em usuários idosos, já que há um envelhecimento da população mundial significante e em um futuro próximo a população de idosos será maior do que a de adultos com condições de cuidar deles, como apontam os dados das Nações Unidas (ONU, 2002). A criação de um sistema que possa auxiliar profissionais de saúde a cuidar de um ou mais idosos é importante, pois em sua maioria esses idosos possuem condições crônicas de saúde e ainda necessitam de compartilhar os profissionais de saúde e os recursos disponíveis. Analisando esse problema, Nunes, Silva e Abrantes (2010) apresentam em seu trabalho os resultados da fase inicial do projeto eCAALYX, que procura melhorar a qualidade

de vida de idosos com as condições discutidas acima e também a comunicação entre os idosos e os responsáveis por cuidarem da saúdes deste idosos.

Nunes, Silva e Abrantes (2010) apresenta uma interface que será manipula através da TV Digital, que auxiliará no processo em estudo pelo projeto eCAALYX. Dois pontos principais investigados no trabalho: (I) a utilização de personas na descrição do processo; e (II) o estudo que mapeia aspectos de mudanças dos usuários pacientes como percepção, cognição, mental e psicológica decorrente da idade.

Para a criação de um sistema com as características comentadas acima, é necessário considerar algumas características específicas dos usuários finais, no caso os idosos, que por muitas vezes não possuem intimidade com as tecnologias mais atuais. A não consideração de tais características pode diminuir o nível de aceitação da ferramenta por parte dos usuários finais. Para identificar as condições especiais de cada usuário e gerar um modelo foi realizado um estudo e direcionado sua completude pela literatura e algumas entrevistas informais realizadas com alguns dos parceiros médicos do projeto, que vivenciam os problemas dos usuários em seu dia-a-dia. Após a coleta das informações foi realizado a criação das personas com base na entrevistas com os médicos e enfermeiras que informarão as necessidades e objetivos para o uso do sistema por seus pacientes. Foram definidos quatro variáveis para representar as personas, nome, foto no ambiente ao qual a persona vive, biografia, e os objetivos e motivações da persona (NUNES; SILVA; ABRANTES, 2010).

As personas criadas entram na fase de requisitos desse projeto para uma melhor compreensão das situações vivenciadas pelos pacientes de cada uma das doenças, escolhidas para tratar no projeto, de maneira a auxiliar o desenvolvimento de conteúdo e adaptações no sistema para cada uma das personas. No projeto eCAALYX, as personas desenvolvidas seguem um modelo com os seguintes atributos: (I) nome; (II) foto; (III) descrição, que possui informações como idade, hábitos de vida e alimentares; (IV) objetivo com o sistema; e (V) motivação com o sistema. Ao todo foram criados 8 personas, sendo que algumas são as principais que descrevem os pacientes e como secundárias os médicos. Os modelos - personas - produzidos foram aprovados pelos médicos que possuem contato direto com o paciente (NUNES; SILVA; ABRANTES, 2010).

Em outro trabalho que ocorreu na universidade de Ohio durante o primeiro ano do curso de engenharia, os professores identificaram a necessidade de criar um projeto para auxiliar seus alunos a terem contato com projetos de engenharia, como se estivessem em uma empresa, auxiliando o entendimento e na conexão entre a teoria e a prática ao longo do curso. Esse projeto possui um fundo social, pois incentivam os alunos a pensarem em produtos que auxiliam o combate da pobreza dos países de terceiro mundo. O projeto tem como objetivo auxiliar os alunos a projetar soluções centradas nos usuários. Nesse tipo de projeto para atender a pobreza dos outros países o desafio dos alunos é maior, pois tiveram pouco ou nenhum contato com essa situação e cultura (ESTELL; REID, 2010).

Um problema enfrentado era permitir que todos os alunos compreendessem as condições dos países em foco sem a necessidade de viajarem para desenvolverem os projetos. Assim, foram criadas personas que representam os habitantes, com base nas observações feitas pelos alunos, que são o público alvo do produto em construção. A escolha de persona ocorreu por tratar-se um modelo detalhado e completo, que auxiliam muito na comunicação das necessidades, objetivos e motivações do usuário. Para criar essas personas, missões de visita são realizadas junto com alguns alunos em países como Quênia e República Dominicana para coletar fotos e relatórios sobre as situações de vida, retratando melhor o usuário final do projeto. Cada nova persona é adicionada à lista de clientes da próxima turma, repetindo o ciclo diversas vezes, mas nenhuma persona é excluída da lista de clientes. Esse projeto auxiliou o desempenho dos alunos durante o decorrer do curso de engenharia, pois ficou mais fácil assimilar as matérias dadas (ESTELL; REID, 2010).

Outra pesquisa estuda a adoção de ferramentas colaborativas em locais de trabalho. Mesmo que esta não seja adota por todas as empresas ou grupos de trabalhos. Quando há um projeto para a criação de ferramentas colaborativas, os projetistas mantém o foco do desenvolvimento em indivíduos, apesar do objetivo da ferramenta ser colaborativa dentro de um grupo de usuários. Esse tipo de abordagem leva as ferramentas a não serem utilizadas por esses grupos como um todo. Para resolver esse problema Matthews et al. (2011) propõem o que chamam de *Collaboration Personas*, que representam grupos de pessoas hipotéticas, onde mantém informações para projetar ferramentas colaborativas, não olhando o comportamento de uma pessoa ou um perfil, mas sim de um grupo de pessoas.

As diferenças apresentadas pelas *Collaboration Personas* com relação às personas tradicionais são: (I) diversos indivíduos inter-relacionados exercendo papéis específicos; (II) foco em objetivos coletivos e elaboração de objetivos individuais que afetam o coletivo; e (III) novo atributo para caracterizar aspectos colaborativos de trabalho do grupo. Assim, *Collaboration Personas* vem como uma ferramenta para auxiliar os projetistas a estabelecerem melhores objetivos para as ferramentas colaborativas (MATTHEWS et al., 2011).

Assim sendo, Matthews et al. (2011) apresentam em seu trabalho, esse novo modelo como uma ferramenta de projeto, que tem como grande objetivo a mudança de foco dos projetistas para análise das necessidades coletivas de um grupo de usuários que interagem entre si dentro de um processo, ao invés das necessidades de um indivíduo apenas, mitigando problemas recorrentes ao projetar ferramentas colaborativas.

Putnam, Kolko e Wood (2012) foram motivados pelo problema de descrever os usuários finais que estão localizados em regiões de desenvolvimento de países diferentes dos projetistas, desenvolvedores e *stakeholders* do projeto. O problema foi vivenciado em dois projetos, um de dispositivos móveis no Kyrgyzstan e o outro de usuários em potencial para dispositivos de armazenamento e tratamento de água em uma região de Andhra Pradesh, na Índia.

Em ambos os projetos, os métodos de análise de requisitos foram feitos com o auxílio da técnica de personas originada em IHC (Interface Humano-Computador), contudo o modelo

da persona utilizada nos projetos teve uma mudança significativa das existentes na literatura de IHC, para atender as necessidades particulares de cada projeto, que estavam voltados a produtos da engenharia civil e eletrônica (PUTNAM; KOLKO; WOOD, 2012).

As personas foram criadas a partir da coleta de dados através de questionários, mas o objetivo da criação foi guiada por uma motivação diferente da persona em sua essência. As personas apresentadas por Putnam, Kolko e Wood (2012), representam não apenas o comportamento do usuário frente a um determinado produto, mas representa também qual o comportamento dele perante a sua rede de relacionamento real, por exemplo, a família e amigos.

Outro atributo importante que se fez necessário adicionar nesse tipo de persona foi um atributo que demonstra como transmitir a diferença cultural e de estilo de vida para a equipe que desenvolve o projeto. Em um dos projetos ainda foi acrescentado um cenário descrevendo como ocorria a interação entre o usuário e o produto. No projeto da Índia preferiu-se adicionar um gráfico com itens importantes sobre a cultura e os tipos de fontes de água existentes e como esse recurso poderia ser armazenado, já que a equipe não estava familiarizada com aquela situação. A partir dos atributos específicos a cada projeto foram criadas as personas para a comunicação entre a equipe (PUTNAM; KOLKO; WOOD, 2012).

Ao fim do processo, Putnam, Kolko e Wood (2012) tinham como objetivo apresentar a possibilidade de utilizar personas não só em IHC, mais também em outras áreas de projetos, para melhorar a comunicação entre os membros da equipe. Para atender esses objetivos são apresentados 5 passos que os autores consideraram essenciais para o processo, sendo eles:

a) Identificar questões chaves que façam sentido à modelagem dos usuários estudados, principalmente relativos à cultura e estilo de vida.

b) Identificar variáveis que possam diferenciar os candidatos de segmento em diversos sentidos e então analisar os dados através do ponto de vista desse candidato.

c) Analisar informações que provenham de outras fontes.

d) Criar uma representação gráfica através de ilustrações enfatizando as diferenças chaves deixando claros os grupos que as personas representam.

e) Criar uma planilha com informações de como a persona foi criada para assegurar transparência com os dados.

Dessa maneira, Putnam, Kolko e Wood (2012) esperam que os projetos possam utilizar a sugestão da persona por eles feita, como uma ferramenta de comunicação efetiva entre usuários, projetistas e *stakeholders* para projetos que possuam esse tipo de diferença cultural e em outros não tão voltados à IHC, mas que a técnica possa ser aplicada como demonstrado no trabalho.

Em outro trabalho, Atzeni et al. (2011) apresentam uma proposta para a criação do *Attacker Personas* que são fundamentadas e validadas sobre uma estrutura de dados e informações sobre invasores de sistema. Este estudo é baseado em projetos de desenvolvimento de sistemas

de segurança, que em geral mantêm o foco em atividades de ameaça e os pontos fracos da ferramenta. Com o foco nos objetivos e características dos invasores foi proposto por Atzeni et al. (2011) utilizar a técnica de personas, originalmente da área de IHC, para criação de um modelo comportamental dos possíveis invasores. Com isso, Atzeni et al. (2011) apresentam uma metodologia para a criação dos *Attacker Personas*, que são as especificações comportamentais dos invasores de sistema. A criação desses difere das presentes na literatura, apenas na fonte de coleta de informações, pois focam em informações passadas por especialistas que enfrentaram situações de ataques em seus sistemas e não são necessariamente os perfis modelados que irão usufruir dos benefícios apresentados ao utilizar personas.

A utilização desse tipo de informação torna a persona menos tendenciosa às crenças dos projetistas do sistema. A fundamentação das *Attacker Personas* é baseada em três características: (I) representação das classes de invasores; (II) representação da convicção criminal para crimes comuns online; e (III) estão contextualizados com o projeto utilizado no estudo de caso do trabalho proposto por Atzeni et al. (2011). O resultado principal apresentado pelos autores foi a criação de um artefato que traz contigo um modelo mais realista para auxilio do desenvolvimento de sistemas de segurança. No futuro pretende-se juntar as técnicas de desenvolvimento de sistemas de segurança e auxiliar as tomadas de decisões na arquitetura das ferramentas de segurança.

Turner e Turner (2011) realizaram uma revisão sobre a aplicação de personas em projetos e trabalhos, e iniciaram um debate com ênfase no carácter psicológico sobre estereótipo e porque é tão presente durante a modelagem de usuários. Ao longo do trabalho notaram que o estereótipo do usuário pode ser preciso e eficiente, contrariando diversos trabalhos inclusive a do criador da técnica de personas. Com esse resultado, apareceram algumas perguntas que ao longo do trabalho os autores tentaram apresentar as devidas explicações.

No estudo de Turner e Turner (2011), é deixado de forma clara que a criação de um modelo de usuário é muito parecida com a criação de um estereótipo, para a grande maioria dos projetistas. No ponto de vista psicológico a criação de um estereótipo é quase inevitável e os autores ainda afirmam que a utilização de personas não resolve esse problema como apontam Cooper, Reimann e Cronin (2007) em seu livro.

Para Turner e Turner (2011) a técnica de personas não difere em nada da técnica de cenários para desenvolvimento de sistemas, exceto ao ponto de vista ao qual a análise é realizada, pois o primeiro tem o foco no ponto de vista do usuário do sistema e o segundo foca nas atividades que são realizadas no sistema.

Ao fim do estudo, Turner e Turner (2011) concluem que a utilização de personas não faz diferença mediante as demais técnicas já utilizadas. Além do mais, a utilização de estereótipos não é ruim ao projeto, dependendo da forma como se aplica a técnica no projeto. Ainda em sua conclusão, os autores recomendam que essa discussão entre a utilização de estereótipos ou personas se encerrassem em seu trabalho, pois como apresentado por eles, no resultado final do projeto ambos possuem o mesmo efeito. Contudo, nessa dissertação acredita-se que a criação

de um esteriótipo pode ser eliminada a partir da automatização do processo, já que dessa forma as possíveis tendências que podem ser geradas são anuladas.

No trabalho de Meissner e Blake (2011), é apresentado um processo de criação de personas para coletar informações culturais para um sistema de tecnologia de informação e comunicação para desenvolvimento (ITCD4D, sigla em inglês). Para aplicar essa técnica os autores realizaram um estudo de caso para a construção de um site em parceria com uma organização não governamental, que tem como objetivo identificar os pontos fracos no sistema de ensino das escolas na África do Sul, que preparam seus estudantes para os desafios da vida.

Na tentativa de um melhor entendimento sobre os usuários deste sistema foram coletados e analisados conhecimentos intermediários ao sistema de uma maneira flexível por influência do processo do projeto. O processo apresentado provém benefícios associados com diferentes personas, como estreitamento de foco, tornar explicito acontecimentos anteriores e implícitos as hipóteses, além de aumentar o conhecimento dos projetistas sobre os usuários dos sistemas. Esse processo também auxilia a estreitar a distância cultural entre o sistema e os usuários (MEISSNER; BLAKE, 2011).

No caso do sistema escolhido por Meissner e Blake (2011) para estudo de caso, a utilização das personas auxiliou no direcionamento dos conteúdos do material oferecido aos alunos ao longo dos cursos de formação, por trata-se de um sistema voltado ao ensino da África do Sul. O resultado obtido foi interessante, pois com o auxílio das personas no direcionamento do material de estudo dos alunos, obteve uma melhora significativa nas notas dos alunos das instituições de ensino no país.

Ao longo dessa seção foram apresentados alguns trabalhos relacionados sobre o tema Personas, e como se pode notar ao longo do texto, a técnica de personas apresentada por Pruitt e Adlin (2005) e Cooper, Reimann e Cronin (2007) é bem flexível e como na maioria dos casos apresentados uma boa opção ao modelar o usuário, por conseguir concentrar as informações sobre o comportamento, necessidade e objetivos do usuário em uma estrutura completa e que torna fácil a comunicação entre equipe de trabalho.

Entretanto, a maioria dos trabalhos apresentados nessa seção são formas de definir a estrutura da persona para armazenar as informações necessárias de acordo com o cenário da aplicação. Apenas um trabalho apresentado procurou definir uma maneira de automatizar a criação das personas (KAN et al., 2010) e ainda assim não conseguiram provar a eficiência do processo, por considerarem que a base de dados com 700 registros seja pequena para esse feito. Ao longo da seção 3 serão apresentados alguns outros trabalhos que utilizam técnicas para automatizar o processo de criação de modelos de usuários, porém a quantidade de trabalhos com esse objetivo ainda é muito baixa.

Alguns trabalhos fora da área de concentração de IHC utilizaram personas como ferramenta para comunicar as necessidades do usuário final dos produtos em desenvolvimento, obtendo sucesso no projeto. Isso demonstra que a técnica de persona é bem flexível e pode ser

aplicada para quaisquer tipos de projetos, como nas áreas de *marketing*, segurança, educação, atendimento ao público, entre outras.

Outro ponto que será trabalhado nessa dissertação é a criação das personas através de um processo e ferramenta que possibilitará a coleta e análise das informações automatizadas, fazendo com que o problema apresentado por Turner e Turner (2011) seja mitigado ou pelo menos seja o primeiro passo para a solução desses problemas. Os autores dizem que não é possível criar uma persona sem que essa seja estereotipada, ou seja, sem que essa possua tendências e semelhanças ao projetista.

Com a flexibilidade apresentada pelo modelo de usuário, persona, obtêm-se detalhes de características do usuário que podem auxiliar na definição dos componentes mais adequados da interface para a persona em análise. Sendo assim, espera-se que a utilização de personas para modelagem de usuário pode ser uma boa opção para automatizar o processo de adaptação de interfaces automatizadas em pesquisas de interfaces adaptativas.

3 CLUSTERING

A utilização de modelos para representação de fenômenos em geral são base para estudos científicos por muitos anos. Para isso, dados experimentais são utilizados como ferramentas para a aplicação desse princípio. Analisar esse tipo de informação em busca de resultados que se aproximem ou possuam um comportamento parecido com o mundo real, muitas vezes gasta-se tempo e é difícil encontrar uma medida plausível (KANTARDZIC, 2011).

Além disso, em certos domínios tais informações são desconhecidas o que dificulta a criação dos modelos, e ainda o estudo desses sistemas em geral é complexo e difícil de serem formalizados matematicamente (KANTARDZIC, 2011).

Com o aumento da utilização de computadores, a quantidade de informações geradas aumentou significativamente nos últimos anos, e com a falta de informações prioritárias para geração de modelos, as informações geradas tornaram-se disponíveis para auxiliar na geração de conhecimento para esses modelos. Contudo, a geração das informações possui um volume alto, o que direcionou o foco dos estudos para formas de armazenamento integra, e não para maneiras de extração de conhecimento (JAIN; MURTY; FLYNN, 1999).

Entretanto, as empresas perceberam que o conhecimento que poderia ser gerado a partir das bases de informação daria uma vantagem competitiva no mercado. O processo que possibilita a extração deste conhecimento armazenado em uma base de informações ou dado é chamado de *Data Mining* (JAIN; MURTY; FLYNN, 1999; KANTARDZIC, 2011).

O processo de *Data Mining* pode apresentar dois objetivos: o primeiro é chamado de predição, que a partir de algumas informações coletadas na base de dados é possível prever o estado futuro de algumas outras variáveis de interesse; o segundo objetivo é chamado de descrição, que analisa os dados em busca de padrões que são traduzidos para uma interpretação mais legível ao ser humano (JAIN; MURTY; FLYNN, 1999).

Uma das técnicas utilizadas em *Data Mining* para obter a descrição dos dados é chamada de *clustering*. A técnica de *clustering* procura identificar em um conjunto finito de dados, algumas características para formação de grupos, onde os elementos pertencentes à um grupo possuem um valor de similaridade igual ou próximo, e quando comparados a outro grupo esse valor de similaridade não existe (KANTARDZIC, 2011).

Para realizar *clustering* é necessária a aplicação de algumas técnicas para associação e classificação dos dados em grupos. Ao aplicar tais técnicas podem-se obter diferentes resultados nos grupos formados ao final do processamento, dependendo da técnica utilizada. Os dados podem apresentar-se das seguintes maneiras entre os grupos (WITTEN; FRANK; HALL, 2011):

a) **Exclusivos**: Neste tipo de *clustering* os elementos pertencem a apenas um grupo e a nenhum outro grupo mais.

b) ***Overlapping***: Para este tipo um elemento pode pertencer a mais de um grupo ao mesmo tempo.

c) **Probabilístico**: Neste caso, um elemento pertence a um determinado grupo com certo grau de probabilidade.

d) **Hierárquico**: Este realiza uma divisão aproximada dos grupos e posteriormente refina até que alcance um resultado que não se altere muito entre as iterações do algoritmo.

Contudo, apesar da existência dos tipos de *clustering* disponíveis, a escolha de qual tipo utilizar, ainda, possui maior influência da ferramenta e dos algoritmos presentes durante a execução de um trabalho, do que propriamente de uma análise e avaliação sobre qual tipo seria o mais adequado frente ao problema em estudo (WITTEN; FRANK; HALL, 2011).

Entretanto, definir qual o melhor método de *clustering* para o problema em estudo é difícil, pois se deve avaliar o método que retorna os grupos que possuem um determinado padrão entre os elementos pertencentes ao grupo, ou seja, os elementos internos, e um padrão diferente aos elementos externos do grupo. Dessa forma é possível segmentar os grupos e deixa-los distintos entre si (MITRA; ACHARYA, 2003).

A utilização de *clustering* é importante na descoberta de padrões sem necessitar de um conjunto de dados para treinamento da ferramenta ou algoritmo, assim é possível descobrir qual elemento pertence a um determinado grupo sem nenhum conhecimento previu, apresentando-se como uma técnica de classificação de dados não supervisionada, como também pode ser encontrado na literatura (MITRA; ACHARYA, 2003).

Clustering é uma técnica que possui aplicações em diversas áreas do mercado. Algumas dessas aplicações podem ser conferidas abaixo (MITRA; ACHARYA, 2003):

a) Reconhecimento de padrões

b) Analise de dados espaciais: possibilitando o mapeamento de regiões geográficas para sistemas de informações especializados.

c) Processamento de imagens: onde é possível segmentar partes da imagem, como por exemplo, o plano de fundo.

d) Computação Multimídia: encontrando objetos que possuam a mesma forma ou coloração em uma base de dados de vídeos, por exemplo.

e) Analise médica: apresentando anormalidades em imagens de ressonância magnética, como exemplo de uma de suas aplicações.

f) Bioinformática: agrupando as assinaturas de genes e DNA.

g) Biométricas: no agrupamento de imagens faciais similares com base em pontos específicos da face.

h) Ciências econômicas: para realizar a analise do mercado.

i) WWW: ao agrupar informações em log com o objetivo de identificar padrões de acesso similares.

j) Interface Humano-Computador: identificando perfis de usuários (AQUINO JUNIOR; FILGUEIRAS, 2005), (AQUINO JUNIOR, 2008), (MASIERO et al., 2011)

Estes são apenas alguns exemplos das grandes áreas de estudo que utilizam *clustering* para auxiliar na análise das informações coletadas, gerando conhecimento sobre cada grupo de dados. Todavia, um dos maiores desafios desta técnica é como realizar analises de dados ou objetos com diferentes tipos de dados, como números, textos e imagens associando-os. Além do mais, estes objetos ainda podem armazenar informações qualitativas e quantitativas, como por exemplo, um perfil de usuário (MITRA; ACHARYA, 2003), (WITTEN; FRANK; HALL, 2011).

A principal medida em *clustering* é o valor de similaridade entre cada elemento ou objeto da base de dados. Esse valor é o que auxiliará no trabalho de classificação e associação dos dados em analise, sendo assim esses valores podem ser representados de diversas formas, de acordo com o problema a ser mitigado. Essas representações são realizadas por posições espaciais, características de conteúdo e distância (MITRA; ACHARYA, 2003) (WITTEN; FRANK; HALL, 2011).

Na seção 3.1 serão apresentados alguns algoritmos utilizados para realização de *clustering* e como são utilizados por estes os valores de similaridades entre os elementos. Dessa forma, é possível identificar qual o melhor algoritmo para cada problema vivenciado pelo especialista e ainda verificar se é possível identificar um único algoritmo aplicável a qualquer problema.

3.1 Algoritmos de *Clustering*

Existem diversos algoritmos que auxiliam no processo de *clustering*. Alguns desses algoritmos são apresentados nessa seção para discussão dos benefícios e problemas que estes proporcionam para a pesquisa apresentada ao longo desse trabalho de dissertação.

Dentre os algoritmos de *clustering*, o k*-means* é um dos mais clássicos e populares entre eles. Para executar esse algoritmo é necessário informar como parâmetro a quantidade de grupos que é desejado, representando o valor de k (WITTEN; FRANK; HALL, 2011).

Com o número de grupos informados o k*-means* escolhe aleatoriamente um número de k pontos, que representarão os centroides do grupo. Após esse passo, todos os objetos são relacionados aos respectivos centroides com base na menor distância euclidiana entre o objeto e a centroide (WITTEN; FRANK; HALL, 2011).

O processo se repete por diversas vezes, até que a centroide se estabilize e alcance o menor erro, que é calculado através dos mínimos quadrados. Para identificar esse critério de parada o valor retornado do centroide e do erro devem se repetir por algumas iterações consecutivas. O método utilizado pelo k-*means* é simples e efetivo, o que faz dele o algoritmo mais popular na tarefa de *clustering* (WITTEN; FRANK; HALL, 2011).

Contudo, o k-*means* como um algoritmo guloso pode convergir para um mínimo local e não para um mínimo global, ficando assim com uma solução que não necessariamente seja a ótima. Fazer com que o k-*means* alcance o objetivo de encontrar a solução ótima é considerado um problema NP-*Hard*, mesmo que o número de grupos escolhidos seja apenas dois (JAIN, 2010).

Apesar do k-*means* possuir algumas vantagens na implementação, ele também possui alguns problemas, como a sensibilidade do resultado final, que dependendo da inicialização dos centroides dos grupos podem levar a resultados bem diferentes entre si. Além disso, a informação do valor ideal para o número de k é uma tarefa complicada, pois dependendo do problema isso não é tão visual ao especialista que está segmentando os dados para efetuar a análise (WITTEN; FRANK; HALL, 2011).

Devido a estes problemas, diversos algoritmos foram desenvolvidos ao longo dos anos visando à mitigação dos problemas apresentados pelo k-*means*. Alguns dos resultados apresentados demonstram melhoras no cálculo da distância realizada pelo k-*means*, além de projetos que executam o algoritmo por diversas vezes e com números de k's diferentes retornando o melhor resultado obtido, que é definido por métodos estatísticos. Esse processo auxilia a determinação de um número k ideal (JAIN, 2010).

Para solucionar o problema de determinar o número de grupos existentes em uma base de dados, Muhlenbach e Lallich (2009) propõem um método de *clustering* baseado na teoria de grafos, chamado de GBC (Graph-based Clustering), que detecta automaticamente o número de grupos existentes na base de dados, sem a necessidade de um parâmetro de *threshold* ou conhecimento prévio da distribuição dos dados.

Este método explora as regiões de influência dos dados construindo grafos entre elas, mantendo-as interligadas. As arestas do grafo que possuem os maiores valores são excluídas e o ponto médio desta aresta determina a divisão da área do grupo contabilizando a quantidade de grupos ou classes automaticamente (MUHLENBACH; LALLICH, 2009).

Durante os testes o GBC apresentou o mesmo resultado no processo de *clustering* que os demais algoritmos, como o k-*means*. Um problema enfrentado pelo GBC é a formação de grupos quando a distribuição dos dados é muito concentrada, pois ele acaba identificando tudo como um grupo único. Para que o resultado do algoritmo seja melhor, os dados devem estar distribuídos de maneira esparsa, caso contrário pode não existir um agrupamento. Outro problema são objetos muito isolados que podem ser identificados como sendo um grupo isolado (MUHLENBACH; LALLICH, 2009).

Muhlenbach e Lallich (2009) ainda aponta que não existe um algoritmo de *clustering* que seja ideal, pois para isso ele não deve receber nenhum parâmetro de *threshold* ou conhecimento prévio, como a quantidade de grupos, para encontrar os grupos ou classes de uma base de dados.

QROCK (*Quick* ROCK) é um algoritmo que foi desenvolvido como uma melhora do algoritmo ROCK que é um algoritmo de *clustering* hierárquico aglomerativo. O QROCK determina a quantidade de grupos através dos componentes conectados como grafos, assim como o GBC. O QROCK apresenta um desempenho computacional maior do que seu antecessor o ROCK (DUTTA; MAHANTA; PUJARI, 2005).

Segundo Dutta, Mahanta e Pujari (2005) a maioria dos algoritmos criados tem enfase em dados numéricos, guiados pelas áreas de estatística e reconhecimento de padrões, e utilizam de diferentes medidas de distâncias para o cálculo dos grupos. Entretanto, a manipulação de dados categóricos é tratada de uma maneira muito natural por estes algoritmos.

Um dos algoritmos que trabalham com dados categóricos é o antecessor ROCK que utiliza uma técnica de dados aglomerativos para agrupar as informações. Para realizar esse agrupamento ele se baseia na ligação realizada entre pares de objetos e o processo aglomerativo auxilia a fundir os grupos terminados ou quando não existem pares entre os grupos, ou ainda se o número de grupos requerido já foi obtido (DUTTA; MAHANTA; PUJARI, 2005).

O algoritmo ROCK serviu como base para o QROCK, que continua o processo de fusão dos grupos até que não reste nenhuma ligação entre eles, restando apenas um grafo de componentes com dados de entradas representados pelos vértices e com dois pontos conectados representando o limite, se o número de ligações entre eles for igual a zero (DUTTA; MAHANTA; PUJARI, 2005).

Para trabalhar com os dados categóricos, Dutta, Mahanta e Pujari (2005) utilizaram o valor de similaridade entre os objetos. O princípio da determinação da similaridade do QROCK é baseado no mesmo utilizado pelo ROCK. A equação 3.1 apresenta o cálculo da similaridade.

$$SIM(X, Y) = \frac{X \cap Y}{X \cup Y} \tag{3.1}$$

A principal diferença entre o ROCK e o QROCK é que o segundo trabalha com um *threshold* θ que representa um valor de similaridade deixando os grupos formados mais naturais, ao contrário de usar um parâmetro que represente o número de grupos. Contudo, Dutta, Mahanta e Pujari (2005) utilizam esse último como um segundo critério de parada do algoritmo.

No geral os resultados do QROCK comparados ao ROCK foram superiores quanto ao desempenho na velocidade de criação dos grupos, e quando comparados os resultados obtidos na geração dos grupos, estes foram iguais aos esperados das bases de dados utilizadas para os testes (DUTTA; MAHANTA; PUJARI, 2005).

Outro algoritmo apresentado que tem como base a densidade dos dados foi apresentado por Ester et al. (1996), onde eles demonstram uma solução para identificar classes ou grupos de dados em base de informações que representam dados espaciais. Um dos problemas encon-

trados nesse tipo de situação são não possuir os conhecimentos mínimos para representar os parâmetros de entrada, identificar os grupos com formas arbitrárias e baixo desempenho com base de dados muito grande.

Visando solucionar esse problema, Ester et al. (1996) apresenta o algoritmo DBSCAN (*Density Based Spacial Clustering of Applications with Noise*) que localiza os grupos e identifica os ruídos nos dados fornecidos, baseado em uma distância mínima entre os pontos e uma quantidade mínima de pontos, que são necessários para definir os dados como um grupo ou classe. Qualquer ponto que não atenda essas características são descartados e considerados como ruídos da base.

Assim, determinar os parâmetros de qual a distância mínima (EPS) e quantos pontos mínimos (MinPts) são necessários para a definição de um grupo é feita através de uma heurística baseada em uma distribuição realizada pelos vizinhos mais próximos que apresenta o melhor resultado nas avaliações das classes ou grupos (ESTER et al., 1996).

O algoritmo apresentado se demonstrou superior ao encontrar formas de classes em uma base de dados e ainda superou o desempenho do algoritmo utilizado como meio de comparação, o CLARANS (NG; HAN, 1994). Contudo, para todos os testes foram utilizados base com informações de pontos. Outras bases que possuem outros tipos de informações, como a base utilizada que possui informações de polígonos, há necessidade de definir uma nova heurística para o algoritmo (ESTER et al., 1996).

Um ponto de atenção do algoritmo DBSCAN é o seu baixo desempenho para dados multidimensionais que precisa de um estudo mais detalhado para provar o caso e melhorar o desempenho do algoritmo (ESTER et al., 1996).

Frey e Dueck (2007) procuram identificar padrões e subconjuntos importantes em processamento de sinais sensoriais. Para realizar essa tarefa utilizando a técnica de *clustering* é possível utilizar um algoritmo que escolhe aleatoriamente os grupos e vai refinando-os a cada iteração do algoritmo.

Contudo, esse método é eficiente somente se a inicialização do algoritmo for próxima do resultado ótimo. Frey e Dueck (2007) apresentam em seu trabalho um método chamado de *Affinity Propagation* que trabalha medidas de similaridades entre pares de pontos.

Esse método enxerga os dados como uma rede de computadores, dessa forma considera todos os pontos como um potencial centroide para os dados. Na sequência são transmitidas informações pelas arestas formadas, verificando o caminho percorrido. Isso ocorre por diversas iterações até aparecer os melhores resultados ao percorrer os caminhos (FREY; DUECK, 2007).

O algoritmo foi utilizado em testes com bases de faces, detecção de genes em dados de micro vetores, identificando trechos de manuscritos e cidades acessadas de forma eficiente por malhas aéreas. Os resultados apresentados durante os testes obtiveram erros menores do que os demais algoritmos que foram comparados, como o k-*means* (FREY; DUECK, 2007).

Apesar de todos os trabalhos apresentados demonstrarem técnicas eficientes de *clustering*, não se pode afirmar que os algoritmos conseguem bons resultados ao trabalhar com

qualquer tipo de dado e problemas, vide tabela 3.1. Ainda é difícil afirmar que os algoritmos apresentam grupos com formatos físicos reais, como por exemplo, circulo ou polígonos, pois os formatos são difíceis de serem definidos pelo especialista, ainda mais em alguns domínios específicos, como por exemplo, modelagem de usuários.

Tabela 3.1 – Tabela comparativa entre os algoritmos.

Categoria	Algoritmo	Parâmetro / Propriedades
clustering hierárquico	Ward	algoritmo aglomerativo
	MST Divisivo	baseado na teoria de grafos
	Clustering Using REpresentatives (CURE)	cada grupo é representado por um conjunto de representações
	RObust Clustering using linKs (ROCK)	$k*$: número de grupos
	QROCK (*Quick* ROCK)	θ: *threshold* de similaridade
hard clustering	*kmeans*	$k*$: número de grupos
clustering baseado em densidade	DBSCAN	ϵ: distância para considerar se 2 pontos são ou não vizinhos
	Affinity Propagation	θ: *threshold* de similaridade
clustering sequencial	*Basic Sequential Algorithm Scheme (BSAS)*	Θ: *threshold* de não similaridade e $k*$: número máximo de grupos

Fonte: Adaptada de Muhlenbach e Lallich (2009).

Pode-se afirmar que um grupo ou classe que pode gerar um bom resultado é aquele que mantém o maior grau de similaridade entre os membros de um mesmo grupo (MITRA; ACHARYA, 2003). Também é importante ressaltar que encontrar uma solução considerada ótima para o problema de *clustering* é considerado como um problema do tipo NP-Completo, pois não é possível solucioná-lo em tempo polinomial (GAREY; JOHNSON, 1990). O grau de similaridade entre os elementos entende-se que deve ser informado e variado de acordo com o interesse do especialista.

Outra questão envolvendo *clustering* é a grande variedade de tipos de dados que podem ser encontradas nos problemas. Para resolver essa necessidade observou-se que a melhor maneira de trabalhar é com uma matriz de similaridade entre os dados, pois essa matriz pode ser calculada de diversas maneiras, estabelecendo apenas uma variável para o algoritmo trabalhar para gerar o resultado.

Sendo assim, o algoritmo de *clustering* quando aplicado em um domínio de modelagem do perfil do usuário, deve se preocupar mais com a qualidade dos dados, do que com a forma que o grupo está assumindo. Quando se trata de características de usuários a quantidade de variáveis é muito grande para representar em um plano cartesiano bidimensional ou tridimensional.

Na seção 3.2 são apresentados os trabalhos relacionados de *clustering* aplicados à problemática de modelagem de usuário que é o foco desse trabalho.

3.2 *Clustering* aplicado na modelagem de usuário

Essa seção tem como objetivo apresentar trabalhos que utilizaram *clustering* com o objetivo de modelar o usuário ou o perfil do usuário. Perfil do usuário é a descrição das características do usuário, retratando os objetivos com base no sistema em desenvolvimento. Já os modelos de usuário descrevem o comportamento, habilidades, relacionamentos, identidade, junto ao sistema. Dessa forma, escolheu-se o modelo de usuário Personas, para representação do usuário, que serão obtidas neste trabalho pela técnica de *clustering* (PRUITT; ADLIN, 2005) (COOPER; REIMANN; CRONIN, 2007) (BARBOSA; SILVA, 2010).

Um trabalho apresentado por Tu et al. (2010) demonstra um processo de criação de personas combinando métodos qualitativos, como observação do usuário e entrevistas, com métodos quantitativos, por exemplo, analise de *cluster*. A técnica de *clustering* tem como objetivo agrupar os usuários mais similares com relação aos objetivos e preferência em tomadas de decisão.

Para desenvolver a pesquisa e criar o processo foi realizado um projeto com uma agência de viagens e conforme solicitado, foram passados pelo departamento de marketing da agência, dois perfis com informações demográficas dos principais clientes. Depois de coletar as informações, Tu et al. (2010) realizaram um pesquisa *online* através de um questionário para identificar os objetivos e tomadas de decisões dos usuários ao utilizar o site da agência. Um total de 24 pessoas respondeu o questionário disponibilizado.

Na sequência definiu-se a dimensão dos dados que compõem as personas e então se aplicou o algoritmo para realizar o processamento de um *clustering* hierárquico, sendo que o algoritmo utilizado foi o *linkage clustering*. Esse algoritmo relaciona perfil a perfil de usuário e vai interligando os mais parecidos como se fossem nós de uma árvore binária até que chegue à raiz, ou seja, definem-se como um único grupo. Dessa forma, o analista pode identificar e definir qual é o melhor número para escolher a quantidade de grupos para o problema em questão. No caso de Tu et al. (2010) optou-se pelo número de 2 grupos, devido à quantidade de perfis informados pela agência conforme o grau de importância.

Com essas informações em mãos as personas são descritas ao longo do processo. As personas criadas auxiliaram o trabalho de reformulação do site da agência e ainda trouxeram um conforto para o projetista do sistema durante o desenvolvimento dessa reformulação. Utilizar a técnica de *clustering* evidenciou as limitações de métodos qualitativos para o problema de criar personas. Assim, para a continuação das pesquisas Tu et al. (2010) pretendem realizar uma comparação na criação de personas utilizando métodos qualitativos contra métodos quantitativos, além de aplicar o método utilizado em uma base de dados maior.

Um novo algoritmo de *clustering* é apresentado para agrupar usuários da internet. O algoritmo apresentado tem como objetivo agrupar os usuários pelo padrão de navegação. Para essa tarefa Xiaoming e Xiaoyan (2009) utilizam as ações de navegação do usuário para re-

presentar as características e um novo método de similaridade para utilizar como medida do algoritmo de *clustering*.

O método utiliza do caminho percorrido em comum e todo o caminho possível para calcular a similaridade entre os usuários e ainda o tempo de permanência no caminho. Esse tipo de trabalho tem como objetivo auxiliar na reformulação hierárquica de um site dentre outras possibilidades (XIAOMING; XIAOYAN, 2009).

Xiaoming e Xiaoyan (2009) declaram que o método desenvolvido funciona. Os testes foram realizados com base nas informações de *log* que possuíam após a coleta durante a navegação dos usuários. Entretanto, o trabalho não demonstra nenhuma prova formal dos resultados, nem comparação com outros métodos embasados em estatística, apenas alguns pontos de informações obtidas analiticamente pelos autores.

Com a intenção de aprimorar a usabilidade do site da biblioteca de sua universidade, Guo e Yan (2011) utilizaram da técnica de personas e classificação da informação através de *card sorting*, procurando deixar o sistema centrado ao usuário.

Para criação das personas utilizou-se métodos qualitativos como questionários, entrevistas e analise dos perfis por parte dos especialistas. Já na parte da arquitetura da informação foi realizado com o *card sorting* e com os resultados apresentados pela técnica, utilizou-se a analise de *clustering* para realizar um estudo melhor dos resultados obtidos. Dessa forma, através de um algoritmo hierárquico é possível obter o mapa do site da biblioteca (GUO; YAN, 2011).

Com toda a reestruturação do site da biblioteca realizada, Guo e Yan (2011) executaram alguns testes de usabilidade para medir a eficiência do trabalho. Os resultados encontrados demonstraram-se eficientes e satisfatórios deixando o site mais limpo e objetivo.

Uma comparação entre métodos quantitativos e qualitativos de *clustering* é apresentada no trabalho de Brickey, Walczak e Burgess (2011). O objetivo de comparar esses métodos é identificar o melhor método para criação de personas para projetos de interface.

Os testes foram realizados com base em quatro métodos para criação de personas, sendo que dois são qualitativos e dois quantitativos. Os métodos qualitativos utilizados foram o processo manual (que é realizado por um profissional da área) e uma técnica chamada de *Latent Semantic Analysis* (LSA). Já os métodos quantitativos utilizados foram a análise fatorial ou análise de componentes principais e análise de *cluster* multivariado (BRICKEY; WALCZAK; BURGESS, 2011).

Para realização dos testes foram realizados alguns questionários disponibilizados *online* e realizado a coleta de informações de log na utilização do sistema. Essas informações foram separadas em texto/verbal para a análise qualitativa e os dados numéricos para as analises através dos métodos quantitativos (BRICKEY; WALCZAK; BURGESS, 2011).

Após os testes, Brickey, Walczak e Burgess (2011) identificaram que o método de análise de componentes principais foi o melhor método dentre todos os apresentados na pesquisa, e outro método quantitativo, a análise de *cluster* demonstrou-se pior que a análise de componentes principais, pelo fato da necessidade de definir o número de grupos que deseja adquirir no

processo. Contudo, ambos os métodos precisam de melhorias para esse desenvolvimento. Os estudos para minimizar esse problema ainda continuam em execução por parte dos autores.

Através do algoritmo de otimização da colônia de formigas, Loyola, Román e Velásquez (2011) apresentam um novo método para analisar o comportamento na internet. O método utiliza das informações coletadas dos comportamentos dos usuários na utilização do site para separa-las em grupos similares através de *clustering*.

Os grupos formados servem como base de treinamento para formigas artificiais, que depois de treinadas essas formigas são soltas dentro de grafos representando a navegação de um site e na sequência as sessões artificiais geradas pelas formigas são analisadas e comparadas com as sessões dos usuários reais do site (LOYOLA; ROMÁN; VELÁSQUEZ, 2011).

Esse método demonstrou-se plausível ao integrar a otimização através da colônia de formigas e técnicas de *web mining*. O resultado de 81% de compatibilidade entre as sessões artificiais em relação as reais foram encontradas através de medidas de similaridades. Loyola, Román e Velásquez (2011) pretendem ainda refinar esse método para conseguir aplicá-lo em diferentes contextos.

Diversos sites permitem que os usuários naveguem sem que seja necessário nenhum método para identificar o usuário. Entender como o usuário utiliza o site e ainda identificar o seu perfil comportamental torna-se uma tarefa difícil, por isso muitos utilizam do recurso de *cookies* para gravar essas informações. Entretanto, esses podem ser apagados pelo usuário o que torna a utilização de *cookies* uma informação ruidosa e com uma vida útil pequena (DASGUPTA et al., 2012).

Com esses problemas em mãos, Dasgupta et al. (2012) apresentam um novo algoritmo de *clustering* para *cookies*, onde é possível gerar perfis de usuários através destes. Os perfis gerados por esse método têm como objetivo identificar características do usuário para recomendar aplicações e propagandas personalizadas ao interesse do usuário.

Para trabalhar com esse problema, foram definidos medidas de similaridade baseada em fatores Bayes (DASGUPTA et al., 2012), formalizando algumas das barreiras encontradas em forma de grafos e ainda utilizaram a aproximação gulosa como heurística e generalização do algoritmo de coloração de grafos. Todavia, ainda é necessário um estudo para melhorar o algoritmo e provar a eficiência teórica e os processos de modelagem de *cookies* como perfil do usuário (DASGUPTA et al., 2012).

Ao analisar os trabalhos apresentados nessa seção pode-se perceber que a utilização de *clustering* para criação de modelos de usuários é muito utilizado, inclusive para personas. Em um trabalho prévio (MASIERO et al., 2011), já foi demonstrado um processo para geração de personas utilizando esta técnica, através do algoritmo *k-means*. Contudo, existem alguns problemas ao utilizar métodos de *clustering* para a criação das personas, com a determinação da quantidade de perfis existente em uma base de dados ou determinar parâmetros que servirão como critérios para o agrupamento dos perfis são alguns dos desafios encontrados nessa área de pesquisa.

Além dos trabalhos apresentados nessa seção existem outros que utilizaram de *clustering* para realizar a modelagem de usuário, como Aquino Junior (2008), Aquino Junior e Filgueiras (2008) e Filgueiras et al. (2005).

Dessa maneira, essa dissertação procura definir um algoritmo que auxilie na criação de personas através do agrupamento de perfis, sem a necessidade de informar qual a quantidade desejada pelo especialista.

3.3 Avaliando o *Clustering*

Por ser um processo geralmente não supervisionado, *clustering* utiliza de alguns métodos para que os resultados possam ser avaliados. Para realizar essa avaliação, normalmente utiliza-se de base de dados bidimensionais, pois são mais fáceis de verificar o resultado já que é possível exibi-lo através de um gráfico, por exemplo (KOVÁCS; LEGÁNY; BABOS, 2005).

As métricas criadas para avaliação dos resultados do processo de *clustering* são utilizadas em conjuntos bem definidos, ou seja, os resultados devem possuir grupos sem sobreposição de elementos (KOVÁCS; LEGÁNY; BABOS, 2005). Três métricas são apresentadas neste trabalho, a variância (KOVÁCS; LEGÁNY; BABOS, 2005), os índices de Dunn (BEZDEK; PAL, 1995) e de Davies-Bouldin (DAVIES; BOULDIN, 1979).

A primeira métrica que será apresentada é a variância. Ela é importante para avaliar a compactação dos grupos gerados e consequentemente a similaridade entre os elementos do grupo. Quanto menor a variância, maior a similaridade entre os elementos dentro do grupo (KOVÁCS; LEGÁNY; BABOS, 2005). O cálculo da variância entre todos os elementos do grupo é dada pela equação 3.2.

Dado um conjunto de grupos G que contém N elementos:

$$\sigma^2 = \frac{1}{G} \sum_{g=1}^{G} \frac{1}{N} \sum_{\substack{i=1 \\ j=i+1}}^{N} (\mu_g - d(i,j)) \tag{3.2}$$

Onde $d(i,j)$ representa a distância entre os elementos i e j, e μ_g é a média das distâncias entre os elementos dentro do grupo. A equação 3.2 realiza não só cálculo da variância de um grupo, mas da variância do conjunto de grupos formados através dos algoritmos de *clustering* gerando um índice para comparação.

Os outros dois índices que são apresentados nessa sessão tem como objetivo medir o quão similar são os elementos dentro de um grupo e ao mesmo tempo o quão diferentes são os elementos em grupos distintos (KOVÁCS; LEGÁNY; BABOS, 2005). Apesar do objetivo dos índices ser o mesmo, os cálculos são diferentes. O índice de Dunn considera a densidade e a

separação dos grupos para avaliar o resultado do algoritmo (BEZDEK; PAL, 1995). O equação 3.3 é utilizada para calcular o índice de Dunn.

$$D = \min_{1 \leq i \leq n} \left\{ \min_{\substack{1 \leq j \leq n \\ i \neq j}} \left\{ \frac{d\left(i,j\right)}{\max_{1 \leq k \leq n} d'\left(k\right)} \right\} \right\} \tag{3.3}$$

Onde $d(i,j)$ representa a distância entre os grupos i e j, e $d'(k)$ é a maior distância interna no grupo k. Quanto maior o índice de Dunn, melhor o resultado do algoritmo.

O último índice apresentado nessa sessão é o de Davies-Bouldin, que considera a separação dos grupos formados para realizar a avaliação do resultado apresentado pelo algoritmo (DAVIES; BOULDIN, 1979). O índice de Davies-Bouldin é calculado através da equação 3.4.

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d\left(c_i, c_j\right)} \right) \tag{3.4}$$

Onde $d(c_i, c_j)$ representa a distância entre as centroides dos grupos c_i e c_j, e σ é a distância média entre todos os elementos dentro do grupo. O melhor valor para o índice de Davies-Bouldin é o menor valor obtido, representando o melhor resultado de agrupamento do algoritmo.

4 Q-SIM: GERANDO PERSONAS COM O VALOR Q

Para resolver o problema discutido ao longo da seção 1, é apresentado nessa seção da dissertação um processo para criação de personas com base em informações coletadas do usuário. O processo de criação é demonstrado na figura 4.1. A proposta de processo, apresentada na figura 4.1, é realizar a captura de determinadas informações do usuário durante a utilização do sistema através de componentes integrados. Os dados capturados são armazenados em uma base de dados, que posteriormente são tratados e aplicados ao algoritmo de *clustering* também proposto por essa dissertação, o *Quality Similarity Clustering* (Q-SIM), fazendo com que as personas sejam criadas ao fim do processo, com base na análise das informações geradas pelo algoritmo.

A definição de um novo algoritmo de *clustering* foi motivada, pois os algoritmos apresentados no capítulo 3 necessitam de informações diversas, como a quantidade desejada de grupos, análise de grupos gerados de maneira hierárquica necessitando da experiência do especialista para definir a melhor formação de grupos para o problema ou limitações dos tipos de dados que os algoritmos foram preparados para trabalhar. Além disso, os algoritmos não garantem que todos os elementos de um grupo possuam um grau mínimo de similaridade entre si, independente dos elementos comparados. Devido a isso, houve a necessidade da definição de uma nova abordagem para identificar grupos de perfis dos usuários que compõe as personas do sistema, de onde originou o algoritmo Q-SIM.

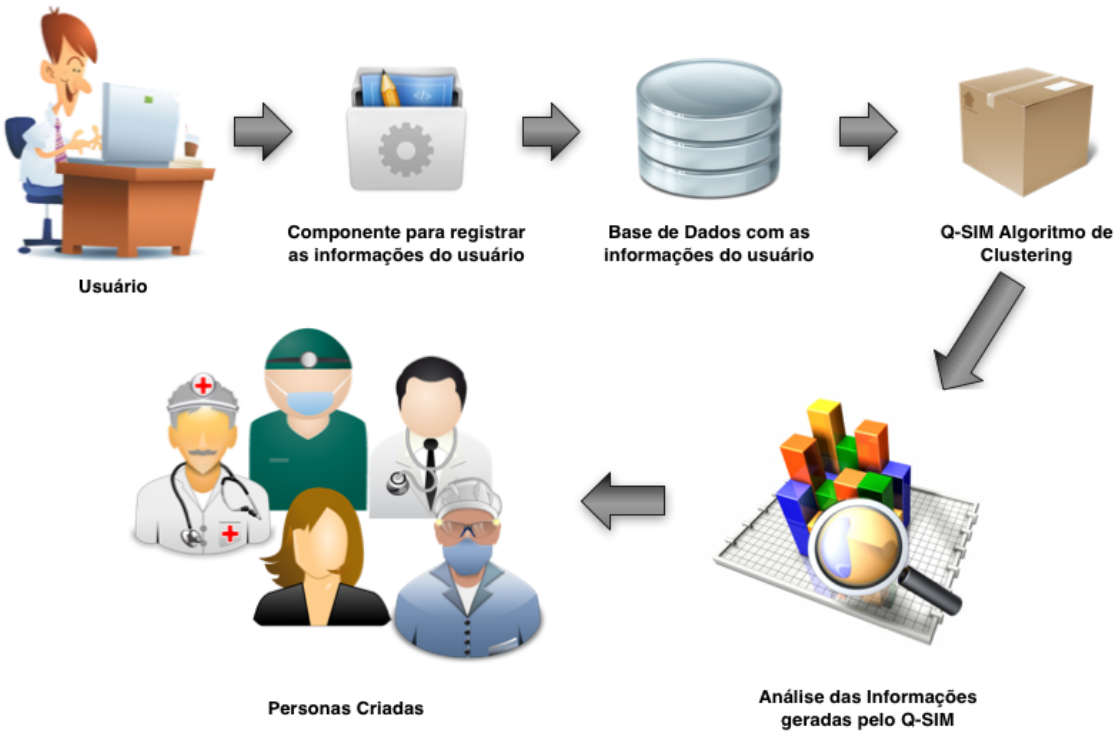


Figura 4.1 – Processo para Criação de Personas de maneira automatizada

O processo apresentado na figura 4.1 possui duas etapas fundamentais: a captura das informações do usuário de maneira automática e o algoritmo Q-SIM. O Q-SIM é o responsável pelo agrupamento das informações dos perfis dos usuários que são a base para a criação das personas. Nas próximas seções serão discutidos o processo do algoritmo e os detalhes de como ele realiza o agrupamento. O componente de captura das informações do usuários será detalhado na aplicação prática do processo no projeto PEAP-PMPT por apresentar características específicas ao projeto. Contudo, vale lembrar que esse componente deve ser modelado de acordo com a necessidade de cada projeto, sendo que restante do processo não sofre alterações na aplicação. Para conferir os detalhes do componente de captura de dados vide seção 6.1.

4.1 Agrupando os perfis de usuário com Q-SIM

Com as informações sobre as características do usuário armazenadas e disponíveis, faz-se necessário executar o algoritmo de *clustering* para que sejam agrupados os perfis de usuários semelhantes para, a partir dos grupos, criar as personas que representaram os usuários do sistema. Contudo, para que esse passo seja realizado, existem dois problemas em particular que precisam ser mitigados para facilitar o trabalho de análise dos perfis.

O primeiro é o número de grupos gerados. Em grande parte dos algoritmos é necessário informar a quantidade de grupos que se deseja. Entretanto essa informação não está disponível ao especialista, dependendo do contexto do projeto, e este deveria realizar uma análise das informações para tentar descobrir essa quantidade. Analisar uma grande quantidade de perfis de usuários torna o processo inviável, transformando o parâmetro do número de grupos uma informação difícil de fornecer. O segundo problema é garantir que todos os membros de um grupo possuam um grau mínimo de similaridade ou semelhança entre si, dessa forma o grupo que será utilizado para criação das personas torna-se homogêneo fortalecendo o valor de representatividade das personas geradas.

Para solucionar esses problemas na modelagem de usuários, criou-se o algoritmo Q-SIM, onde este recebe a informação de um parâmetro: o grau de similaridade. A partir do parâmetro, são criados os grupos, sendo que todos os elementos devem manter o grau de similaridade como o valor mínimo entre si. Assim, é possível variar o grau de similaridade e dessa forma conhecer a quantidade de grupos existentes para um desses valores, e ainda conhecer o tamanho dos grupos de acordo com o valor da similaridade. A base do Q-SIM teve como motivação a teoria de casos relacionados (*related sets*) (SMYTH; MCKENNA, 2001), utilizada na técnica de raciocínio baseado em casos. Os detalhes do algoritmo serão apresentados mais adiante na subseção 4.1.3.

O agrupamento é realizado com base no valor de similaridade entre os elementos. Esse valor é calculado de acordo com a regra de similaridade que é definida na implementação do

Q-SIM, que calcula a matriz de similaridade dos elementos, neste caso os perfis dos usuários. As regras para o cálculo da matriz de similaridade serão apresentadas na subseção 4.1.1. Outro ponto importante é a normalização dos valores de similaridade que deve ficar entre 0 e 1, sendo 1 totalmente similar e 0 o inverso. A normalização dos valores é apresentada na subseção 4.1.2. Com o cálculo da matriz realizado o algoritmo Q-SIM determina os grupos existentes na base de dados para a criação das personas ao fim do processo.

4.1.1 Cálculo da Similaridade

Devido a diversidade dos tipos de variáveis existentes, como categóricas (informações textuais) ou numéricas, os algoritmos de *clustering* utilizam a medida de similaridade entre os objetos para identificar os padrões existentes na base de dados e determinar os grupos com base nesses padrões. Esse tipo de decisão ocorre pois a medida de similaridade consegue representar melhor essa diversidade de informações, como se pode observar no trabalho apresentado por Dutta, Mahanta e Pujari (2005) no capítulo 3.

Existem diversas formas para realizar o cálculo do valor de similaridade. Quando possuímos apenas dados numéricos em um elemento representado em um espaço, por exemplo, podemos utilizar o cálculo da distância entre eles para determinar a similaridade. Dentre as distâncias existentes na literatura temos a distância euclidiana como a mais comum entre elas, representada na equação 4.1 (DEZA; DEZA, 2009). Outras distâncias podem ser utilizadas para esse cálculo, como a distância de Manhattan (4.2 (BLACK, 2004)) ou a Mahalanobis (4.3 (MAHALANOBIS, 1936)), onde essa última é mais aplicada a dados matriciais, entre outras distâncias.

$$Sim(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i + Y_i)^2} \tag{4.1}$$

Aonde X e Y apresentados na equação 4.1 são os vetores que contém as informações das coordenadas de cada objeto.

$$Sim(X,Y) = \sum_{i=1}^{n} |X_i - Y_i| \tag{4.2}$$

Aonde X e Y apresentados na equação 4.2 são os vetores que contém as informações das coordenadas de cada objeto.

$$Sim(\vec{X},\vec{Y}) = \sqrt{\left(\vec{X} - \vec{Y}\right)^T \cdot S^{-1} \cdot \left(\vec{X} - \vec{Y}\right)} \tag{4.3}$$

Aonde S apresentado na equação 4.3 é a matriz de covariância das informações.

Os métodos apresentados acima são utilizados com maior frequência para variáveis numéricas, porém esses métodos também podem ser aplicados a variáveis categóricas, convertendo-as em códigos numéricos. Contudo, utilizar esse procedimento não é adequado, pois essas variáveis geralmente não possuem razão, ou seja, não existe uma hierarquia definida entre os

valores. Para tais tipos de variáveis utilizam-se alguns outros métodos, como o Qui-Quadrado (LATTIN; CARROL; GREEN, 2011) ou a divisão da interseção das características dos objetos pela união dos mesmos, conforme apresentado na equação 3.1. O método da equação 3.1 é utilizado quando você quer comparar um conjunto de dados categóricos, porém pode-se realizar uma variação dessa equação para comparação de caracteres podendo assim comparar uma variável categórica por vez.

Outra forma de realizar o cálculo da similaridade é utilizando um método que combine os dois tipos de variáveis, categóricas e numéricas, para o cálculo da similaridade entre os objetos, no nosso caso os perfis de usuários. Para esse tipo de cálculo utiliza-se um método aplicado em raciocínio baseado em casos, onde é calculado o valor de similaridade para cada característica ou variável do objeto (similaridade local) e posteriormente o valor de similaridade para o objeto como um todo (similaridade global).

Para o cálculo da similaridade global, pode-se estabelecer alguns pesos entre as características do objeto de tal forma, que as mais importantes para definir a similaridade entre os objetos ganhe uma maior importância ao longo do processo. Assim, uma soma-produto entre as similaridades locais e os pesos determinados para as características definem o cálculo da similaridade global, conforme demonstrado na equação 4.4.

$$Sim(X, Y) = \frac{\sum W_i \cdot sim\left(X_i, Y_i\right)}{\sum W_i} \tag{4.4}$$

O cálculo a similaridade local, ainda pode-se utilizar do seguinte método demonstrado na equação 4.5, além de todos os outros métodos apresentados ao longo dessa subseção.

$$sim(X_i, Y_i) = 1 - \left(\frac{|X_i - Y_i|}{(\max - \min)}\right) \tag{4.5}$$

Essa dissertação utilizará o último método apresentado, que combina o cálculo da similaridade local e global, para determinar a matriz de similaridade entre todos os perfis de usuários coletados através do componente apresentado na seção 6.1. A escolha deste método ocorre já que as características de perfis de usuário possuem diferentes tipos de dados e informações.

4.1.2 Normalização dos Dados

O passo da normalização das informações ou dados é importante para padronizar a escala de valores entre as variáveis numéricas. Em geral, a normalização é realizada para manter os valores entre 0 e 1 (LATTIN; CARROL; GREEN, 2011). A forma mais simples de realizar uma normalização é aplicar a equação 4.6, que divide o valor da característica do objeto pelo valor máximo encontrado entre os objetos, para essa característica analisada.

$$X_{i_{normalizado}} = \frac{X_i}{\max_{X_i}} \tag{4.6}$$

Entretanto utilizar a equação 4.6 para normalizar os dados, pode gerar uma tendência ou generalização nas informações. Pode existir uma concentração nos dados em um determinado intervalo generalizando a informação coletada. Esse fenômeno ocorre principalmente quando a média dos dados não é a melhor medida de dispersão a ser utilizada para identificar o ponto médio. Por exemplo, um conjunto de idade dos usuários $\mathcal{I} = \{10, 18, 19, 20, 20, 18, 20\}$. No conjunto $\mathcal{I}$ o valor 10 cria uma tendência nos dados e a aplicação da equação 4.6 resultaria em uma generalização. Para situações desse tipo, a equação 4.7 resulta em uma normalização que melhor representa os valores reais.

$$X_{i_{normalizado}} = \frac{X_i - \min_{X_i}}{\max_{X_i} - \min_{X_i}} \tag{4.7}$$

Para verificar se a média não está tendenciosa a algum tipo de ruído, precisa-se garantir que o desvio padrão dos dados, seja no máximo 30% da média ($\sigma \leq 0.3 \cdot \mu$). Caso o valor de σ seja maior do que isso os dados estão tendenciosos devido a um ruído existente. Ao ocorrer esse tipo de situação é necessário determinar um valor de corte para o máximo e outro para o mínimo da variável (LATTIN; CARROL; GREEN, 2011). Assim, as escalas e quantidade da amostra são mantidas com uma distribuição uniforme mesmo após a normalização.

4.1.3 Algoritmo Q-SIM

Preparação dos dados concluída, nesse momento existe a necessidade de agrupar os perfis de usuário que são similares. Contudo, o objetivo é criar grupos de perfis de usuários que possuam uma maior similaridade entre si. Para isso, o primeiro passo do Q-SIM é definir o *Related Set* para cada um dos perfis baseados na similaridade desejada pelo especialista. O *Related Set* de um perfil é determinado através de um grupo de perfis que atende no mínimo o valor Q de similaridade do perfil alvo p. A definição formal é, adaptado de Smyth e McKenna (2001):

Definição 1. *(Related Set) Um Related Set de um perfil alvo p, denotado por $RS(p)$, é um grupo de perfis formados pela seguinte fórmula:*

$$RS(p \in \mathcal{P}) = \{\forall q \in \mathcal{P} / similaridade\,(p, q) \geq Q\}$$

Aonde:

- *$\mathcal{P}$ é o conjunto universo dos perfis coletados.*
- *Q é o valor de similaridade entre 0 e 1.*

Note que p é incluso em seu próprio *Related Set*, pois $similaridade\,(p, p) = 1$.

Cada *Related Set* possui uma quantidade de perfis que tem um valor de similaridade Q em relação ao perfil alvo p. Embora todos os perfis contidos em um *Related Set* sejam similares ao perfil p, não há garantia que um perfil q seja similar ao perfil r, sendo que $q, r \in RS\,(p)$. Dessa forma, procura-se um subconjunto de $RS\,(p)$, onde esse subconjunto atenda o valor

mínimo de Q entre todos os elementos pertencentes à ele. Esse subconjunto é chamado de *Reduced Related Set*, definido a seguir.

Definição 2. *(Reduced Related Set) Um Reduced Related Set de um perfil alvo p, denotado por $RRS(p)$, é um grupo de perfis formados pela seguinte fórmula:*

$$RRS(p) = \{\{c_1 \dots c_n\} \in RS(p)/similaridade(c_i, c_j) \geq Q, 1 \leq i \leq n, 1 \leq j \leq n\}$$

Nota-se que existem diversos subconjuntos $RRS(p) \in RS(p)$, onde este grupo seja formado apenas por perfis que sejam similares a todos os perfis existentes no $RRS(p)$. Entretanto, procura-se o maior $RRS(p) \in RS(p)$. Com o intuito de encontrar o maior *Reduced Related Set*, define-se um processo que inicia-se com a busca de um perfil $q_i \in RS(p)$ que possua a Maior Interseção Comum (MIC) definida a seguir.

Definição 3. *(MIC) Dado $RS(p) \in \mathcal{P}$ que contém os perfis $\{q_1, q_2, \dots, q_n\}$, a maior interseção comum deste RS é denotada por:*

$$MIC(p) = q_i \in RS(p) | RS(p) \cap RS(q_i) :$$
$$max(RS(p) \cap RS(q_1), RS(p) \cap RS(q_2), \dots, RS(p) \cap RS(q_n))$$

Com base no MIC (definição 3), o próximo passo do Q-SIM é buscar pelo Maior Grupo Similar (MGS). Esse processo é recursivo e procura atender a definição a seguir.

Definição 4. *(MGS) Dado $RS(p) \in \mathcal{P}$, o maior grupo similar entre o perfil alvo p e os perfis $\{q_1, q_2, \dots, q_n\} \in RS(p)$ é denotado pela seguinte fórmula:*

$$\mathcal{T}_0 = RS(p) \cap RS(MIC(p))$$
$$\mathcal{T}_1 = RS(\mathcal{T}_0) \cap RS(MIC(\mathcal{T}_0))$$
$$\forall \mathcal{T}_n \neq \emptyset : \mathcal{T}_n = RS(\mathcal{T}_{n-1}) \cap RS(MIC(\mathcal{T}_{n-1}))$$
$$MGS(p) = p \cup MIC(p) \cup MIC(\mathcal{T}_0) \cup MIC(\mathcal{T}_1) \cup \dots \cup MIC(\mathcal{T}_n)$$

O melhor RRS é aquele que garante o maior número de perfis do *Related Set* original. Contudo, escolher perfis de um determinado grupo para maximizar o número de perfis dentro de um RRS é difícil de computar e necessita de um algoritmo com um grande tempo de processamento. Como uma solução, implementa-se o algoritmo guloso utilizando o processo da função MGS que adota-se como o RRS do perfil p. O processo encontra um conjunto subótimo para o $RRS(p)$.

Nesse momento do processo de agrupamento, cada perfil possui seu próprio *Reduced Related Set*. Obviamente, existem muitas interseções entre todos RRS's desde que cada perfil possua um RRS e também sejam membro de diversos outros RRS's pertencentes a outros perfis. A união de todos os RRS's forma o conjunto universo dos perfis existentes $\mathcal{P}$. Cada RRS é, portanto, um subconjunto de $\mathcal{P}$.

Próximo passo é encontrar o menor número de subconjuntos $RRS \in \mathcal{P}$, cujo todos os membros estejam em $\mathcal{P}$. Um conjunto de RRS's que contém todos os perfis de $\mathcal{P}$ é chamado

de $\mathcal{C}$. O problema de encontrar o menor $\mathcal{C}$ é conhecido como *set-cover* e é provado ser NP-Completo (GAREY; JOHNSON, 1990). Uma solução aproximada é também um algoritmo guloso que seleciona o RRS, onde este englobe o maior número de perfis, ainda não escolhido. É uma boa aproximação para o problema de *set-cover* e também provê uma boa solução, próxima da ótima.

Escolher o RRS com o maior número de perfis para compor um dos grupos de $\mathcal{C}$ não é suficiente. Para esse passo é necessário encontrar uma métrica que escolha além do RRS com maior número de elementos àquele que contém os perfis mais concentrados ou próximos entre si. Com esse objetivo, uma função de densidade é definida. A função de densidade utilizada pelo Q-SIM considera o RRS com o maior número de perfis e também o com perfis mais próximos. O cálculo da densidade do RRS utiliza como base o coeficiente de variação dos dados, que mede a proximidade entre os perfis. A definição da função de densidade é dada a seguir.

Definição 5. *(Função Densidade) A densidade de $RRS(p)$ é calculada pela seguinte fórmula:*

$$densidade(RRS(p)) = \frac{tamanho(RRS(p))}{\frac{\sigma(RRS(p))}{\mu(RRS(p))}}$$

Aonde:

- $\sigma(RRS(p))$ *é o desvio padrão das similaridades dos perfis pertencentes ao* $RRS(p)$.
- $\mu(RRS(p))$ *é a média das similaridades dos perfis pertencentes ao* $RRS(p)$.

Obs.: Caso $\frac{\sigma(RRS(p))}{\mu(RRS(p))}$ *seja igual a zero, então a densidade será definida apenas pelo* $tamanho(RRS(p))$.

Calculado a densidade de todos subconjuntos $RRS \in \mathcal{P}$, conforme a definição 5, o Q-SIM seleciona o RRS que possui maior valor de densidade para torna-se parte de $\mathcal{C}$. Neste processo todos os perfis em $\mathcal{C}$ não são considerados nas próximas seleções. Devido a isso, quando um RRS é eleito como parte de $\mathcal{C}$ os valores de densidades são recalculados para todos os $RRS \in \mathcal{P}$ excluindo qualquer elemento pertencente a $\mathcal{C}$.

A partir da escolha de um $RRS \in \mathcal{P}$, através da função de densidade, o Q-SIM verifica se os perfis no RRS, ainda não pertencentes a $\mathcal{C}$, não podem ser incluídos nos grupos já existentes de $\mathcal{C}$. Para essa tarefa utiliza-se o algoritmo 4.1 apresentado a seguir nesta dissertação. É importante garantir o valor Q entre todos os elementos do grupo.

Após a execução da função 4.1, verifica se todos os perfis não escolhidos foram alocados em algum dos grupos existentes em $\mathcal{C}$. Caso essa condição seja verdadeira, não há a necessidade de executar os próximos passos, permanecendo na escolha do RRS com maior valor de densidade até que pelo menos um perfil não escolhido permaneça neste estado após a função 4.1.

Dado a seleção de um RRS, chamado A de um perfil alvo p_1, e um conjunto de grupos $\mathcal{C}$:

function inserirNovosPerfis(A)
Ordene $\mathcal{C}$ de forma decrescente pela $C_i \cap A$
$B = perfisNaoEscolhidos(A)$
Enquanto $B\ != \emptyset$ **faça**
  Para cada $p \in B$ **faça**
    Para cada $C \in \mathcal{C}$ **faça**
      Para cada $q \in C$ **faça**
        Se $similaridade(p, q) \geq Q$ **Então**
          C = C + p;
          B = B - p;

Algoritmo 4.1 – Função para inserir os perfis não escolhidos nos grupos existentes

Quando pelo menos um perfil de um determinado RRS permanece não escolhido após a função 4.1 este RRS torna-se um grupo do conjunto $\mathcal{C}$, porém existe diversas interseções entre ele e os demais grupos já existentes. Para resolver o problema das interseções é necessário a criação de dois ou mais conjuntos independentes.

Definição 6. *(Conjuntos Independentes) Dois conjuntos $\mathcal{A}$ e $\mathcal{B}$ são considerados independentes se $\mathcal{A} \cap \mathcal{B} = \emptyset$.*

A definição 6 é uma premissa de grupos, aonde não há interseção entre eles. A separação dos grupos é realizada com base em suas centróides (definição 7). O algoritmo 4.2 demonstrado a seguir transformam dois conjuntos em conjuntos independentes. É importante a percepção de que cada RRS está relacionado a um perfil p. O perfil p do RRS é chamado de perfil alvo do conjunto, como mencionado anteriormente.

Definição 7. *(Centróide) Dado um conjunto de características $\{c_1 \dots c_m\}$ pertencentes a um perfil p, e um grupo de n perfis denotado por $\mathcal{A}$, onde $p \in \mathcal{A}$. A centróide de $\mathcal{A}$, que contém as características $\{k_1 \dots k_m\}$, é definida pela fórmula:*

$$\forall p\ \in \mathcal{A}, \forall k\ \exists\ Centroide(\mathcal{A}), k_i = \sum_{j=1}^{n} \frac{p_j\ (k_i)}{n}, 1 \leq i \leq m$$

O processo realizado pelo algoritmo 4.2 pode ser observado na figura 4.2, onde: (I) identifica-se os grupos que possuem interseção; (II) calcula-se as centróides de cada grupo; (III) compara-se a similaridade do perfil contido na interseção com as centróides; e (IV) aloca-se o perfil no grupo ao qual a centróide for mais similar a ele.

Com os conceitos e funções envolvendo o algoritmo de agrupamento por similaridade Q-SIM apresentados, a última parte do algoritmo, definição dos grupos, é discutida na sequência dessa dissertação.

Os grupos podem ser encontrados escolhendo o menor número de RRS independentes que englobem todos os perfis pertencentes a $\mathcal{P}$. Considerando a aproximação gulosa do processo, os grupos são definidos pelo algoritmo 4.3, apresentado a seguir.

Dado dois RRS, um chamado de A de um perfil alvo p_1 e outro chamado B de outro perfil p_2:

function criarConjuntosIndependentes(A,B)
Inicialize C = $A \cap B$;

Para cada $p \in C$ **faça**
  Se $Similaridade(p, Centroide(A)) \geq Similaridade(p, Centroide(B))$ **Então**
    B = B - p;
  Senão
    A = A - p;

Algoritmo 4.2 – Função para criação de conjuntos independentes

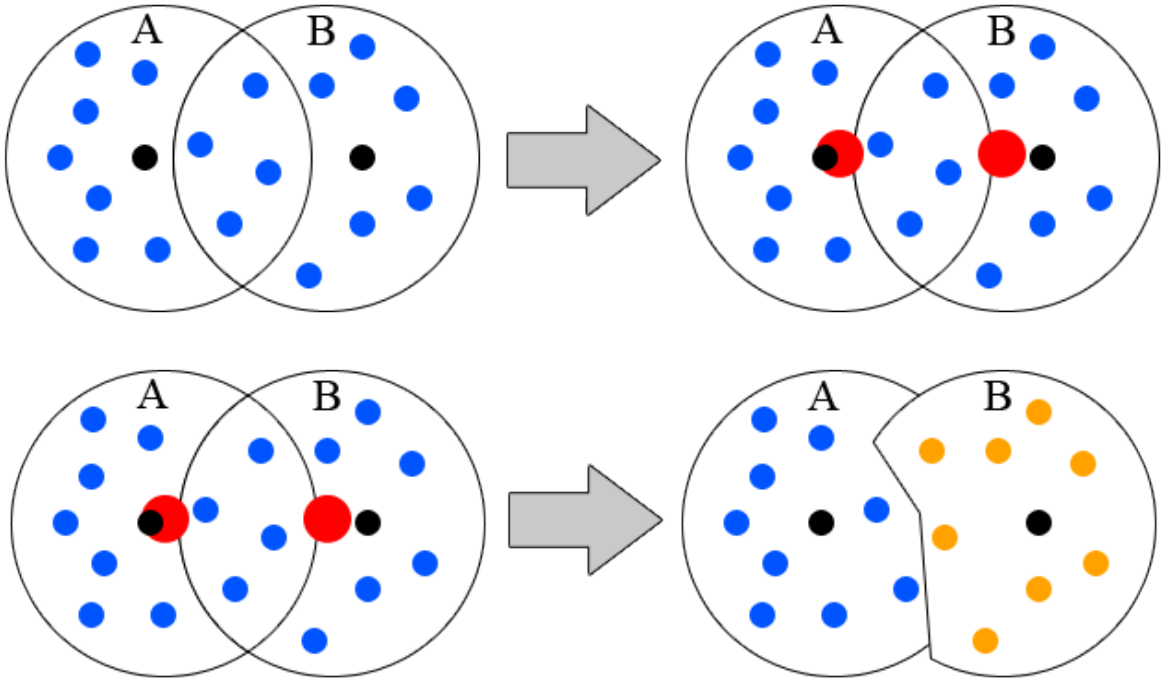


Figura 4.2 – Resultado obtido através do algoritmo 4.2 para os grupos A e B. Os pontos vermelhos representam as centróides calculadas de cada grupo

A formação dos grupos finaliza quando todos os perfis de $\mathcal{P}$ também estão em $\mathcal{C}$. Com os grupos formados, o Q-SIM ainda realiza mais dois processos com o intuito de minimizar o número de grupos formados e suavizar os limites de cada um dos grupos formados. Ambos processos são definições gulosas, obtendo uma solução subótima. Para suavizar as bordas do grupos, deve-se comparar todos os perfis $p_i \in \mathcal{C}$ com as centróides de cada grupo existente e realocando o perfil no grupo ao qual a centróide é mais similar a ele, sem ferir o valor Q.

O segundo processo para minimizar é a união de dois ou mais grupos existentes, tornando-se apenas um. Para que isso seja possível é necessário que a definição 8 seja atendida, e na sequência verificar se a união dos grupos em análise não prejudicam o valor Q entre os elementos desse grupo.

Dado um grupo de RRS chamado R de cada perfil $p \in \mathcal{P}$:

function definirGrupos(R)
Inicialize C = $\emptyset$;
Inicialize S = $\emptyset$;
Repita
  Selecione um perfil $p \in (\mathcal{P} - C)$ com $\max(densidade(RRS(p)) \in (\mathcal{P} - C))$;
  Adicione todos perfis $\in RRS(p)$ em C;
  AUX = inserirNovosPerfis($RRS(p)$);
  Se AUX ! $= \emptyset$ **Então**
    Adicione $RRS(p)$ para o conjunto de grupos S;
    Para cada $q \in S$ **faça**
      criarConjuntosIndependentes(q,$RRS(p)$);
Até que $(\mathcal{P} - C) == \emptyset$

O Resultado é um conjunto de grupos em S. Cada conjunto $\in S$ é um grupo de perfis utilizados para formar as personas.

Algoritmo 4.3 – Função para definição dos grupos

Definição 8. *(Possível União de Grupos) Dois conjuntos $\mathcal{A}$ e $\mathcal{B}$ são possíveis de união se $\forall p \in \mathcal{A}$ e $\forall q \in \mathcal{B} \mid similaridade(p, q) \geq Q$.*

Dessa forma, o processo do algoritmo Q-SIM está completo e possível de gerar o menor número de grupos com o valor Q de qualidade entre cada um dos elementos dos grupos. O processo completo pode ser visualizado através da figura 4.3.

Portanto, o algoritmo Q-SIM, apresentado na figura 4.3, realiza o seguinte processo: Calcula a similaridade entre todos os perfis existentes na base de dados (passo 1) com base em uma regra de similaridade, como por exemplo as apresentadas na subseção 4.1.1. O resultado do passo 1 é uma matriz de similaridade. A partir da matriz de similaridade são calculados os *Related Sets* dos perfis utilizando a definição 1 (passo 3), gerando a matriz de *Related Sets*.

Na sequência, utilizando a matriz de *Related Sets*, o algoritmo Q-SIM realiza o cálculo pelo maior *Reduced Related Set* (definição 2) de cada perfil, utilizando definição 4 no passo 5 do processo. A resultante deste passo é a matriz de *Reduced Related Sets* (RRS). Com os RRS definidos é possível calcular, no passo 7, a densidade de cada um através da definição 5. O valor da densidade é importante para selecionar o perfil que iniciará a composição dos grupos (passo 9).

No passo 10, verifica-se se existem outros grupos selecionados, caso essa resposta seja negativa retorna ao passo 9, recalculando a densidade dos RRS excluindo os perfis pertencentes ao primeiro grupo. Em caso positivo, o Q-SIM procura inserir os perfis ainda não agrupados entre os grupos existentes (passo 11), sempre mantendo o valor Q entre os elementos do grupo. Se todos os perfis forem realocados entre os grupos existentes o algoritmo retorna ao passo 9, recalculando a densidade dos RRS excluindo os perfis já agrupados (passo 12).

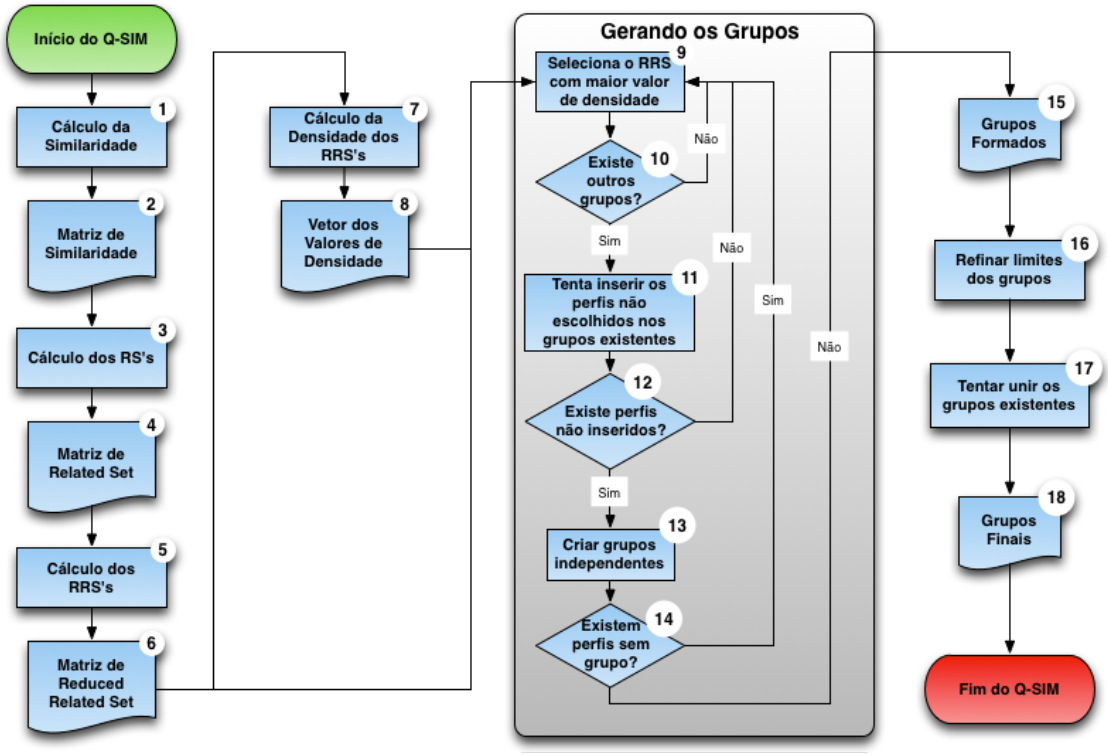


Figura 4.3 – Processo completo do Q-SIM

Quando no passo 12, ainda existirem perfis sem grupos, o algoritmo define o RSS do perfil selecionado como um grupo do processo de *clustering*. Contudo, existe intersecções entre os grupos existentes e essas devem ser resolvidas. Para isso, utiliza-se o processo da definição 6 que gera os grupos independentes (passo 13). Ao fim desse passo, caso exista ainda perfis sem grupos, o algoritmo retorna ao passo 9, recalculando a densidade dos RRS excluindo os perfis já agrupados.

Em caso contrário, os grupos já estão formados porém, busca-se o menor número de grupos e a maior similaridade intra grupo. Para auxiliar nesse objetivo, dois outros processos são realizados nos próximos passos, antes de apresentar os grupos finais. No passo 16, o Q-SIM realiza a comparação de todos os perfis existentes para saber, com base na centróide dos grupos, qual é o grupo mais similar e realoca esse perfil caso ele mantenha o valor Q entre os demais elementos do grupo. E no último passo do Q-SIM, o 17, é realizado um processo para tentar unir os grupos baseados na definição 8.

Ao final do processo, os grupos finais são apresentados pelo Q-SIM. Com as informações produzidas pelo Q-SIM é possível gerar as personas para o sistema em análise. Na seção 4.2 serão apresentados os métodos para criar as personas.

4.2 Obtendo as Personas

A partir deste ponto, os grupos de perfis de usuários estão formados e é necessário de alguns procedimentos para que as características mais representativas no grupo sejam escolhidas para formar a persona que o representará. Os procedimentos que são adotados nessa dissertação já foram apresentados no trabalho de Masiero et al. (2011), onde é apresentado um processo de criação de personas que utiliza operações para encontrar a média (4.8), a mediana (4.9) ou a moda das informações contidas dentro dos grupos, definindo assim cada variável das personas.

$$media = \frac{\sum_{i=1}^{n} X_i}{n} \tag{4.8}$$

$$mediana = \frac{(n+1)}{2} \tag{4.9}$$

As operações devem ser aplicadas em cada uma das características ou variáveis dos grupos de perfil. Assim, é possível definir um valor em comum para cada variável dos perfis existentes nos grupos. Antes de se aplicar algumas das operações precisa-se primeiro identificar o tipo de variável que a característica adota. Para variáveis categóricas (variáveis textuais) recomenda-se utilizar a moda, pois essa apresenta como resultado o valor com maior frequência dentro do intervalo de valores existentes no grupo. Uma segunda opção para dados categóricos é atribuir códigos para cada um dos possíveis valores encontrados e a partir desse ponto aplicar qualquer uma das três operações, conforme realizado no trabalho de Masiero et al. (2011). Essa última opção só é recomendada a aplicação caso esses valores possuam uma razão, caso contrário deve-se descartar-lá.

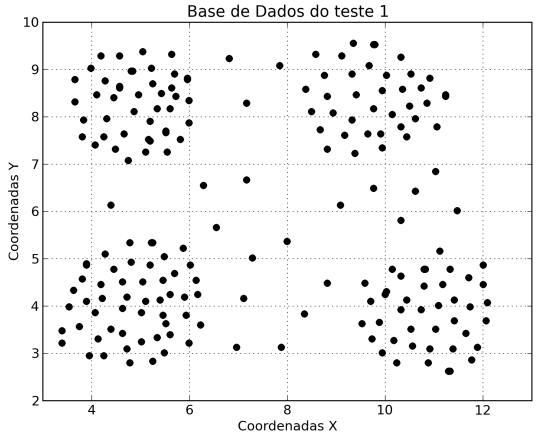
Para variáveis numéricas, pode-se aplicar qualquer uma das operações já que em teoria todas essas operações retornam o mesmo resultado. Entretanto, dentro dos valores das variáveis pode existir em algum dos perfis um valor que deixe a média dos dados tendenciosa. Um exemplo para esse cenário, imagine valores da idade dos usuários, representada pelo vetor $I = \{20, 19, 20, 18, 19, 58\}$. A idade de 58 anos, faz com que o valor da média da idade seja igual a 25 anos, aproximadamente. Contudo, esse valor não corresponde a média da idade da população, pois existe o ruído de uma idade muito alta.

Nesse caso, recomenda-se utilizar a moda ou a mediana como valor em comum para o grupo. Um método de verificar se a média serve como medida para obter o valor da característica para o grupo é verificar se essa atende ao seguinte critério $\sigma \leq 0.3 \cdot \mu$, onde σ é o desvio padrão dos dados e μ é a média. Se essa condição for verdadeira, significa que a média pode ser utilizada como a operação para obter o valor desejado. A partir desse ponto, obtêm-se os valores em comum das variáveis para o grupo. Basta o especialista realizar a análise dessas informações, e inserir dentro da descrição da persona as informações obtidas através desse conhecimento gerado com a técnica de *clustering*.

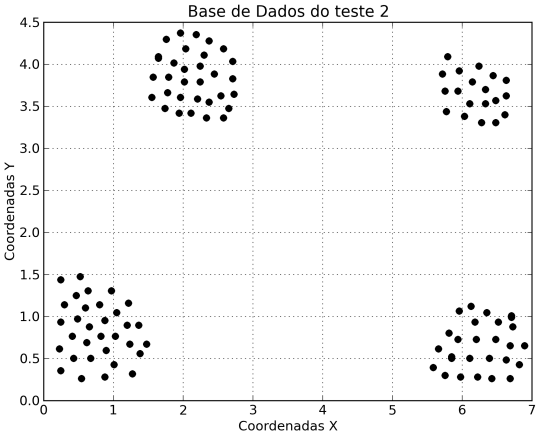
Aplicando esse método para construir efetivamente as personas, atinge-se o objetivo de representar um grupo de perfis de usuário através de uma persona. Dessa forma, as personas obtidas podem ser publicadas dentro de um projeto de interface e/ou sistema com o objetivo de auxiliar na construção de um produto centrado ao usuário, ou sua continua melhoria.

5 AVALIANDO O Q-SIM

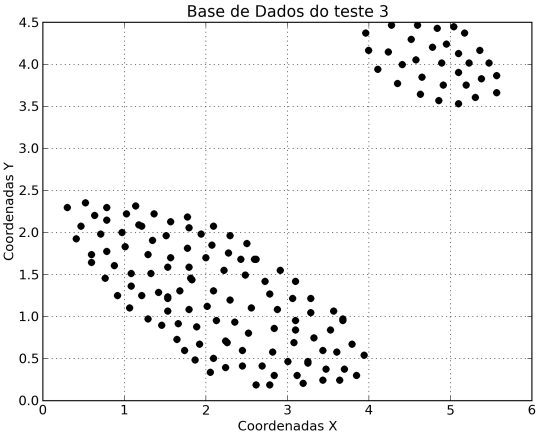
Foram necessários dois passos para a conclusão do trabalho, a validação e a aplicação do Q-SIM. Para realizar a validação do Q-SIM foram construídas quatro bases de dados, vide figura 5.1, com informações bidimensionais de pontos no espaço cartesiano. A utilização de dados bidimensionais facilitam a interpretação, visualização e avaliação do comportamento e resultado do algoritmo (LEGáNY; JUHáSZ; BABOS, 2006). Para realizar as comparações de resultados do Q-SIM, selecionou-se os algoritmos *k-means* (WITTEN; FRANK; HALL, 2011), DBSCAN (ESTER et al., 1996) e *Affinity Propagation* (FREY; DUECK, 2007) que possuem o mesmo princípio do Q-SIM.



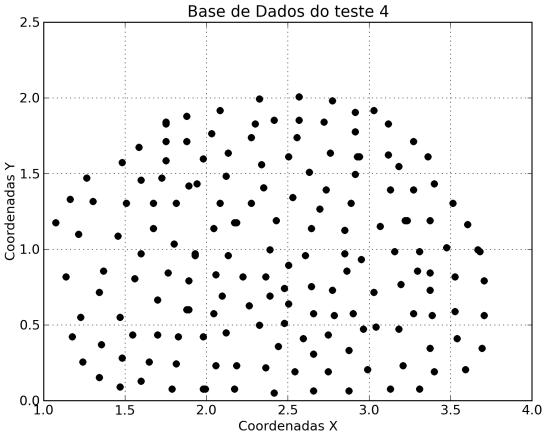
(a) Base de Dados 1



(b) Base de Dados 2



(c) Base de Dados 3



(d) Base de Dados 4

Figura 5.1 – Bases de dados utilizadas nos testes de validação do algoritmo Q-SIM

As bases de dados construídas possuem características especiais, as quais os algoritmos encontram dificuldades em realizar o trabalho. A base 5.1a, possui quatro grupos sólidos com ruídos entre esses grupos. Esse tipo de situação pode atrapalhar a identificação dos grupos, principalmente quando existe a necessidade de incluir esses ruídos entre os grupos. A base 5.1b

possui apenas os quatro grupos sólidos sem nenhum ruído entre eles. Essa é a base de dados mais fácil para um algoritmo classificar os grupos.

Já a base 5.1c, possui um conjunto denso e grande de dados e um segundo conjunto menor, que simula as duas situações extremas de *clustering*, dados esparsos e aglomerados, em uma base única. Por fim, na base de dados 5.1d existe um aglomerado de informações, tornando difícil a identificação dos grupos.

Realizada a comparação dos resultados entre os algoritmos, aplica-se o Q-SIM na base de dados coletada através do sistema PEAD-PMPT, para a criação dos personas do sistema identificando as habilidades dos usuários. Na sequência do texto serão discutidos os resultados apresentados pelos quatro algoritmos em cada uma das bases de validação, apresentando e discutindo a quantidade de grupos formada, alguns índices de validação para *clustering*, e outros. Por fim, as personas obtidas com o Q-SIM no sistema PEAD-PMPT serão apresentadas.

Com o intuito de ilustrar os passos dos testes realizados, a figura 5.2 apresenta os passos da validação do Q-SIM junto aos demais algoritmos.

A metodologia apresentada na figura 5.2, realiza os seguintes passos: (I) As bases de dados são inseridas em cada um dos algoritmos em análise; (II) Os algoritmos produzem os grupos para cada uma das bases de dados utilizadas nos testes; (III) Os agrupamentos obtidos através dos algoritmos são submetidos as análises do especialista e as métricas estatísticas apresentadas na seção 3.3; e (IV) Por fim, os resultados são apresentados. Dessa forma, nessa dissertação aplica-se a metodologia da figura 5.2 para validação do Q-SIM.

Para realização dos testes foi necessário identificar quais os parâmetros de *threshold* de cada um dos algoritmos, ou seja, os valores limiares utilizados na definição dos grupos. Como os parâmetros de *threshold* entre os algoritmos são diferentes, padroniza-se a entrada destes para que simulem o mesmo resultado obtido com o parâmetro de entrada Q do Q-SIM.

No caso do k-*means*, como é necessário escolher o número de grupos desejados, utilizou-se como entrada a mesma quantidade de grupos gerada pelo Q-SIM. O DBSCAN necessita de dois parâmetros, o número de indivíduos mínimo para a formação do grupo e a distância para determinar o limite entre os grupos. Optou-se então, por utilizar 1 para o mínimo de indivíduos e o valor Q atribuído ao Q-SIM para o *threshold* de limite entre os grupos, simulando exatamente o mesmo comportamento do Q-SIM. Por fim, no *Affinity Propagation* não foi necessário determinar qual o parâmetro, já que este se comporta como uma rede neural do tipo *Competitive Learning*, então não necessita de um *threshold* que define a quantidade de grupos ou um limite máximo entre os grupos.

Iniciou-se os testes com a base de dados 5.1a, que possui um total de 195 pontos, representando 4 grupos definidos com alguns ruídos entre cada um destes. O tipo de base representado pela figura 5.1a torna difícil o trabalho do algoritmo de identificar os grupos devido aos ruídos entre cada um deles, porém no caso de modelagem de usuário, deve-se considerar esses ruídos no agrupamento. Para o teste com essa base utilizou-se o valor Q igual a 0.6, que é o

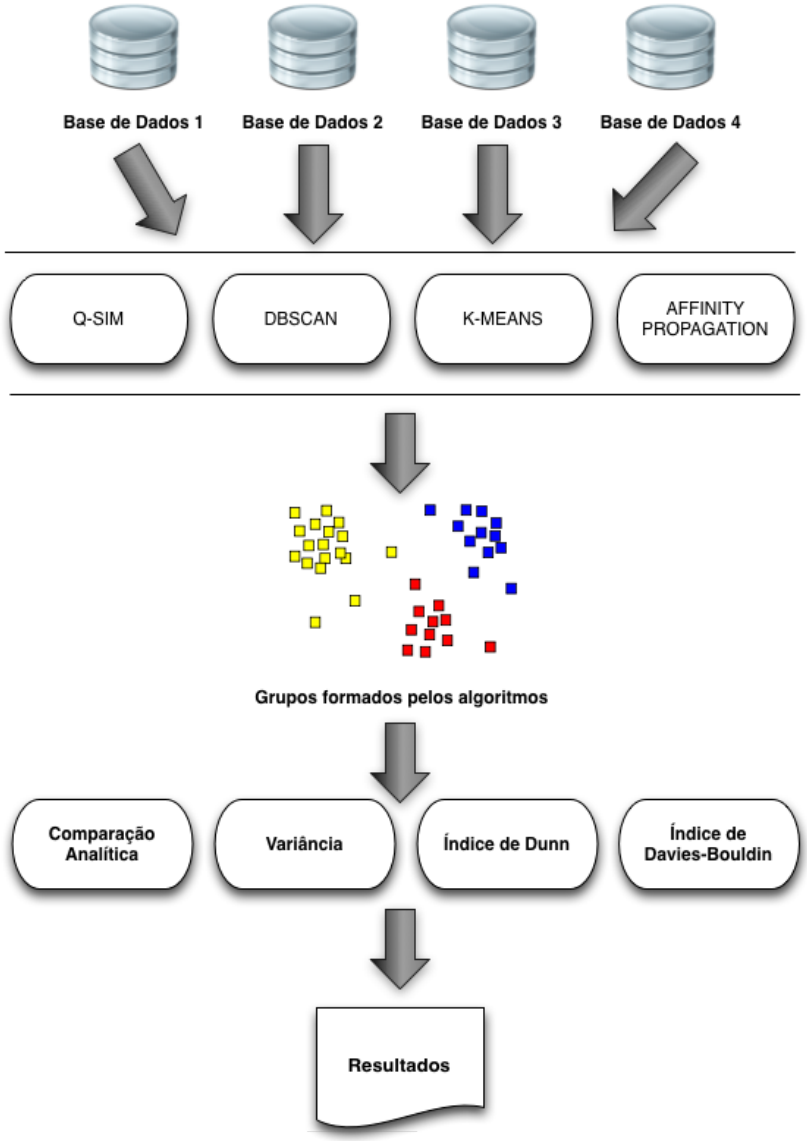


Figura 5.2 – Metodologia para validação do algoritmo Q-SIM

equivalente a 60% de similaridade. A figura 5.3 apresenta o resultado do Q-SIM para a base de dados 5.1a.

O Q-SIM apresentou um total de seis grupos neste primeiro teste. Observa-se que os dois grupos a direita ficaram maiores englobando dois dos quatro grupos bem definidos, quando se remove os ruídos entre esses grupos. Já a esquerda é apresentado quatro grupos ao todo. Esse resultado é esperado justamente pela existência maior dos ruídos próximos aos grupos gerados. Apesar dos ruídos, o resultado obtido pelo Q-SIM é satisfatório já que ele manteve os elementos dentro de cada um dos grupos formados atendendo o valor Q na qualidade entre as similaridades entre os perfis.

Para iniciar as comparações para a base de dados 5.1a, executou-se o k-*means* e como entrada do parâmetro de número de grupos foi inserido o número seis, que foi a quantidade de grupos encontrada pelo Q-SIM. Como a inicialização do k-*means* é aleatória foram executados

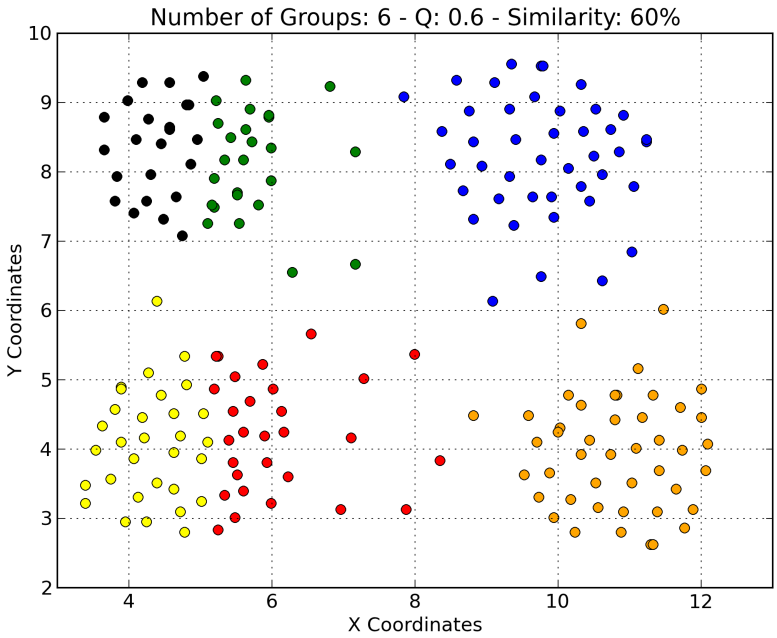
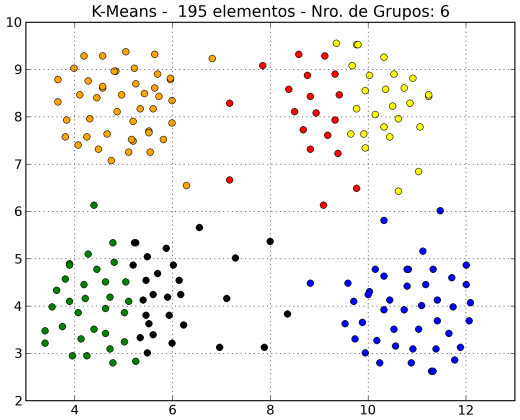
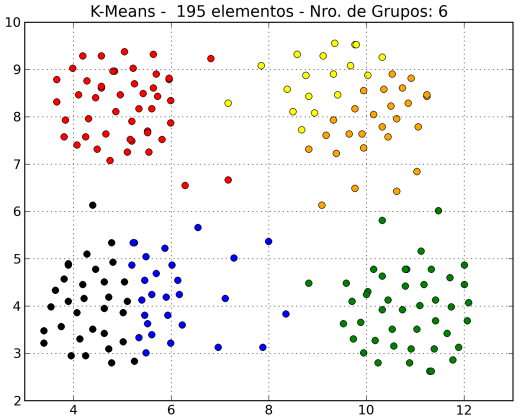


Figura 5.3 – Resultado obtido através do Q-SIM para a base de dados 1

um total de dez testes para o algoritmo. A figura 5.4 apresenta dois resultados obtidos ao longo dos testes.



(a) Primeira execução do algoritmo



(b) Segunda execução do algoritmo

Figura 5.4 – Resultado obtido através do *k-means* para a base de dados 1

Os resultados apresentados pelo *k-means* foram bem próximos aos resultados do Q-SIM. Contudo, nota-se algumas características que podem prejudicar os limites dos grupos apresentados pelo *k-means*, principalmente na segunda execução representada na figura 5.4b. Um dos problemas apresentados é em relação a manter a qualidade dos grupos gerados atendendo uma similaridade de no máximo 60%. Esse problema é perceptível no grupo vermelho da execução 5.4b, onde os três ruídos acabam ferindo a qualidade. Para atender essa similaridade máxima é necessário aumentar o número de grupos, o que nos leva contra o objetivo de encontrar o menor

número de grupos com maior qualidade entre seus elementos. Isso faz com que o Q-SIM nesse ponto seja melhor que o k-*means*.

Outro ponto relevante existente na segunda execução do k-*means* são os limites entre os grupos, por exemplo, entre os grupos amarelo e vermelho que possui um ponto amarelo entre os vermelhos (na área de ruído) que poderia ser absorvido pelo grupo vermelho. Quanto a essa observação o Q-SIM apresentou limites mais suaves para a base 5.1a.

O segundo algoritmo a ser comparado ao Q-SIM é o DBSCAN. Como há a necessidade de informar dois parâmetros para a execução do algoritmo, o número mínimo de elementos e a distância máxima de similaridade entre dois pontos, indicou-se o valor de um para o primeiro parâmetro, já que o Q-SIM não faz esse tipo de distinção, e para o segundo parâmetro optou-se por utilizar o mesmo valor que Q (0.6). Esses valores foram inseridos na entrada do algoritmo para que o comportamento do DBSCAN fosse o mesmo que o do Q-SIM. A figura 5.5 apresenta o resultado do teste com o DBSCAN.

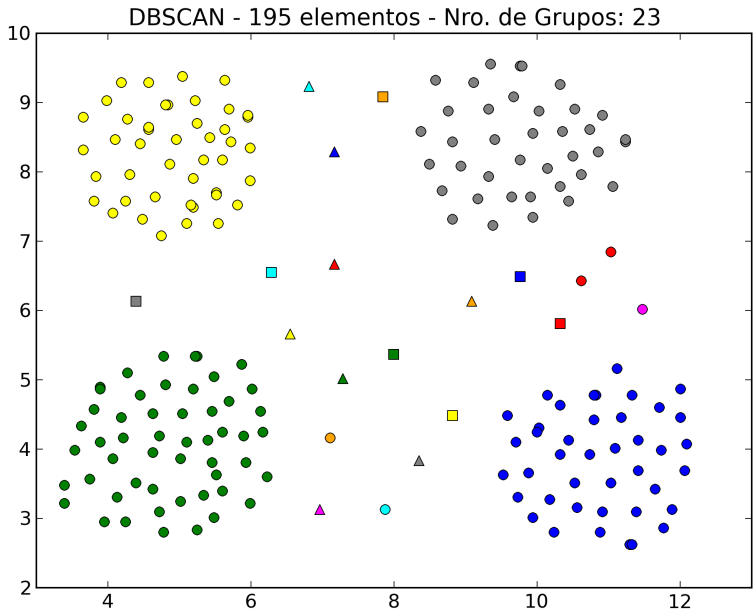


Figura 5.5 – Resultado obtido através do DBSCAN para a base de dados 1

O DBSCAN apresentou como resultado um total de 23 grupos encontrados. O algoritmo identificou com exatidão os quatro grupos existentes na base de dados, entretanto os 20 pontos existentes que representam os ruídos entre os grupos foram identificados como praticamente um grupo cada ponto. Esse tipo de resultado para modelagem de usuário é prejudicial, uma vez que deseja-se que esses ruídos sejam inclusos nos grupos formados. Além disso, grupos que possuem um ou dois perfis no máximo gera o mesmo trabalho para o especialista, que analisar cada usuário de forma individual, fugindo da proposta da técnica de personas.

Na identificação dos 4 grupos de maneira exata o DBSCAN foi o algoritmo que apresentou o melhor resultado, mas no tratamento dos ruídos para o problema de modelagem de usuário

ele não teve um comportamento esperado. Para os ruídos só existiram duas opções, tratar como casos individuais aumento em grande escala o número de grupos e especializando muito uma análise por parte do projetista ou excluído os casos especiais da análise. Nesse ponto, tanto o Q-SIM quanto o *k-means* apresentaram um resultado melhor que o DBSCAN.

O último algoritmo para comparação com o Q-SIM é o *Affinity Propagation*. Como não existem parâmetros de *threshold* para determinar a quantidade e o formato dos grupos, esse algoritmo foi apenas executado nas bases de validação. O resultado apresentado pelo *Affinity Propagation* é demonstrado na figura 5.6.

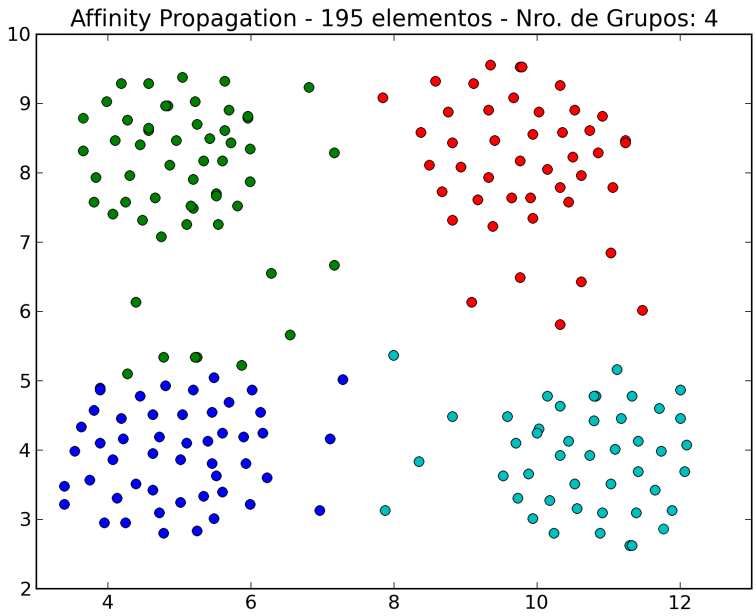


Figura 5.6 – Resultado obtido através do *Affinity Propagation* para a base de dados 1

Foram 4 grupos gerados pelo *Affinity Propagation* na base de dados 5.1a. Foi o menor número de grupos encontrado entre os algoritmos utilizados nessa dissertação sendo esse um ponto positivo para o *Affinity Propagation*. Entretanto os grupos gerados por este algoritmo tem um característica que o coloca em desvantagem para com os outros, ele tenta agrupar todos os elementos em grupos com o maior número de elementos possíveis e isso faz com que ele gere apenas quatro grupos, para essa base de dados. Esse comportamento leva o *Affinity Propagation* a não manter o grau de similaridade homogêneo entre seus grupos.

Assim, concluí-se as análises visuais dos algoritmos. Entretanto existem alguns índices que auxiliam na avaliação dos grupos criados, conforme apresentado no capítulo 3. Para essa análise, foi necessário algumas considerações quanto aos algoritmos devido as particularidades de cada um. O Q-SIM, DBSCAN e *Affinity Propagation* como não possuem uma aleatoriedade na sua execução, foram obtidos os índices apresentados na sequência com apenas uma execução dos algoritmos. Já o *k-means* foi executado 10 testes e foram extraídos os melhores e os piores índices de cada um e a discussão será com base nos resultados gerados.

O primeiro dos índices utilizado foi o *Dunn Index* (BEZDEK; PAL, 1995) que procura averiguar o quão próximo são os elementos intra-grupo e o quão distante estão os elementos extra-grupo. Quanto maior o índice melhor é o resultado obtido pelo algoritmo. A figura 5.7 apresenta os índices de Dunn para a base de dados 5.1a.

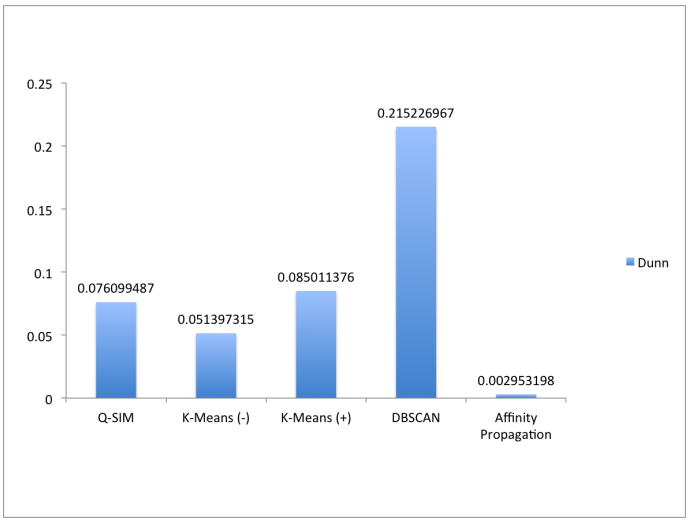


Figura 5.7 – *Dunn Index* para a base de validação 1

Pelo índice de Dunn, o melhor algoritmo DBSCAN apresentou o melhor resultado, porém esse resultado foi influenciado pela quantidade de grupos que possuem apenas um elemento fazendo com que o índice fique cada vez maior. Ao analisar os demais algoritmos nota-se que o *Affinity Propagation* apresentou o pior índice, pois tentou gerar grupos muito grandes o que desbalanceou a qualidade de seus grupos. O *k-means* em seu pior caso ficou bem abaixo dos outros algoritmos, sendo melhor apenas que o *Affinity Propagation*.

Já em seu melhor caso o *k-means* superou as expectativas e seria o melhor índice apresentado, se não fosse o resultado do DBSCAN. Contudo, a sua média para esse índice em 10 execuções é de 0.067 o que o torna o segundo pior índice obtido. Dessa forma, o Q-SIM, por ser mais estável que os demais, demonstrou-se a melhor opção para essa base segundo o índice de Dunn, unindo o menor número de grupo com mais elementos similares.

O segundo índice utilizado para mensurar a qualidade dos grupos gerados pelos algoritmos é o índice de Davies-Bouldin (DAVIES; BOULDIN, 1979) que, assim como o índice de Dunn, mede o quão similares são os elementos de um grupo e o quão diferentes são os elementos de grupos distintos. Só que diferentemente do índice de Dunn, o índice de Davies-Bouldin é melhor para os valores menores encontrados. A figura 5.8 apresenta dos índices de Davies-Bouldin para a base de dados 5.1a.

Novamente o DBSCAN mostrou-se com o melhor índice. Isso ocorreu devido as medidas dos grupos serem pequenas por causa dos grupos formados de apenas um elemento. Como o índice de Davies-Bouldin maximiza o seu resultado dividindo a soma das média de similaridades dos grupos formados pela distância entre suas centroides, o DBSCAN foi favorecido devido a sua distribuição. O *Affinity Propagation* obteve o segundo melhor índice que é justifi-

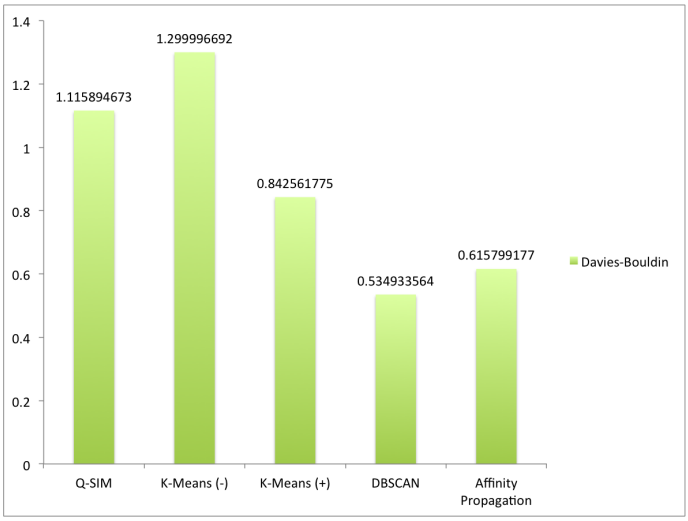


Figura 5.8 – *Davies-Bouldin Index* para a base de validação 1

cado pelo fato dos grupos serem maiores e em menor quantidade do que os demais algoritmos. Contudo, deve-se lembrar que ele não manteve uma similaridade de pelo menos 60% entre seus elementos.

Nesse teste, o *k-means* obteve bom resultado para o seu melhor caso, sendo o terceiro colocado na classificação entre os algoritmos. Mas o seu pior caso foi muito ruim, a variação de seus resultados obtiveram um desvio padrão de 0.14 aproximadamente, e a média dos resultados foi de 1.035 o que faz com que o Q-SIM para o índice de Davies-Bouldin seja o pior dos algoritmos para essa base de dados.

Como última medida de avaliação para os algoritmos, utilizou-se a variância das similaridades entre os elementos do grupo. Essa medida tem como objetivo medir o quão compacto ou similar são os elementos do grupo. Quanto menor a variância, maior a similaridade e a compactação dos grupos, sendo assim o menor resultado das variâncias é considerado o melhor resultado para a avaliação. A figura 5.9 apresenta os resultados dos algoritmos em relação a compactação dos grupos obtidos.

Para a medida da compactação dos grupos o DBSCAN obteve o pior índice na classificação, devido aos grupos com apenas um elemento não foi possível o cálculo desse índice. O *Affinity Propagation* atingiu um índice satisfatório, mas não foi o melhor, ele ficou classificado como o segundo pior para esse quesito.

O *k-means* apresentou uma oscilação entre seus resultados. Seu pior caso ficou melhor do que o *Affinity Propagation* e seu melhor caso foi a melhor compactação obtida entre os algoritmos. Contudo, a média desse índice foi de 0.0046 o que torna o Q-SIM o melhor resultado entre os quatro algoritmos. Sendo assim para a avaliação dessa base de dados, vide figura 5.1a, o Q-SIM demonstrou-se a melhor opção, ficando abaixo das expectativas apenas no índice de Davies-Bouldin. A consolidação dos resultados é apresentada na tabela 5.1

A escolha apontada pelos resultados, como o algoritmo para a base de dados 5.1a é o Q-SIM. Apesar de não ter sido o melhor algoritmo, de acordo com os índices de Dunn e

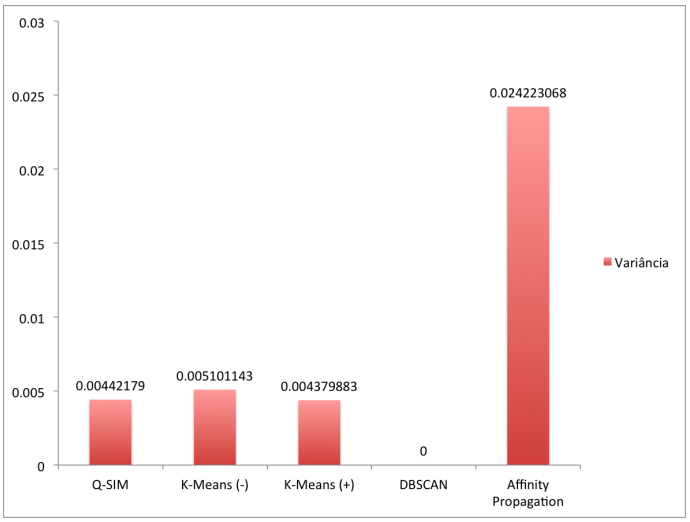


Figura 5.9 – Variância para a base de validação 1

Tabela 5.1 – Resultados dos Algoritmos Vs. Análises para Base de Dados 5.1a

Algoritmo	Analítico	Variância	Dunn	Davies-Bouldin
Q-SIM	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆	⋆
k-means	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆
DBSCAN	⋆	⋆	⋆⋆	⋆⋆⋆⋆⋆
Affinity Propagation	⋆⋆	⋆	⋆⋆⋆⋆⋆	⋆⋆

Davies-Bouldin, o agrupamento realizado pelo Q-SIM foi o que apresentou o melhor fator de similaridade entre os elementos (variância), conforme na tabela 5.1, e analiticamente os limites entre os grupos ficaram bem definidos e sólidos.

Com um total de 110 pontos, a segunda base de dados (figura 5.1b) foi criada representando quatro grupos definidos sem nenhum ruído entre eles. Base de dados que possui a característica de grupos esparsos, em geral, não apresentam muita dificuldade para os algoritmos em identificar os elementos similares e agrupá-los. Para os testes nesta base de dados, também definiu-se um valor Q igual a 0.6, representando a similaridade de 60%. A figura 5.10 apresenta o resultado obtido pelo Q-SIM ao executar o agrupamento em cima destes pontos.

Conforme esperado, o algoritmo Q-SIM identificou com sucesso os quatro grupos existentes mantendo a similaridade entre os elementos de um mesmo grupo com pelo menos 60%. Da mesma maneira que o teste da primeira base, foi informado ao *k-means* o interesse por encontrar quatro grupos, quantidade de grupos encontrados pelo Q-SIM. O resultado do *k-means* para essa base de dados é apresentado na figura 5.11, onde essa possui o resultado de duas execuções do algoritmo devido sua aleatoriedade nos resultados.

Assim como o Q-SIM, o resultado apresentado pelo *k-means* ocorreu como esperado, já que este identificou os quatro grupos existentes sem grandes dificuldades. Mesmo nas diversas execuções praticamente todos os resultados obtidos foram iguais, conforme demonstrado nas figuras 5.11a e 5.11b.

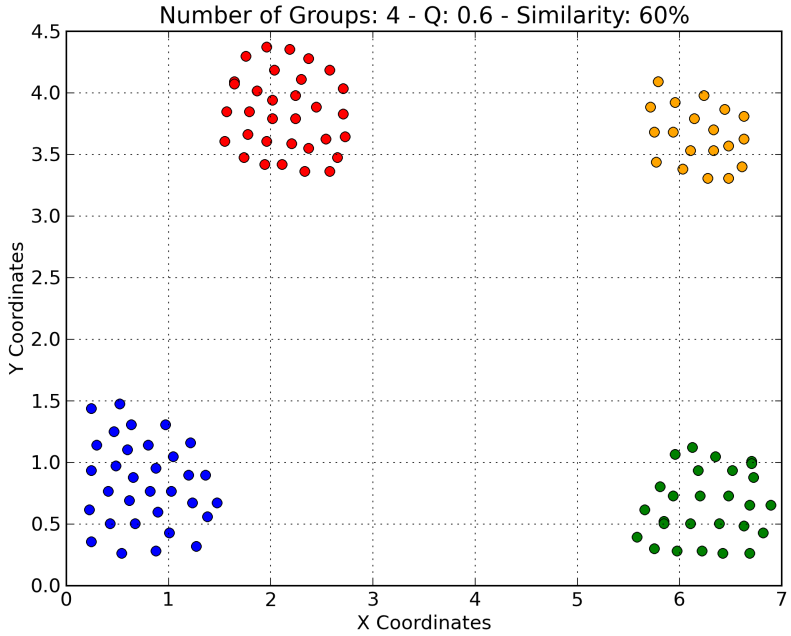
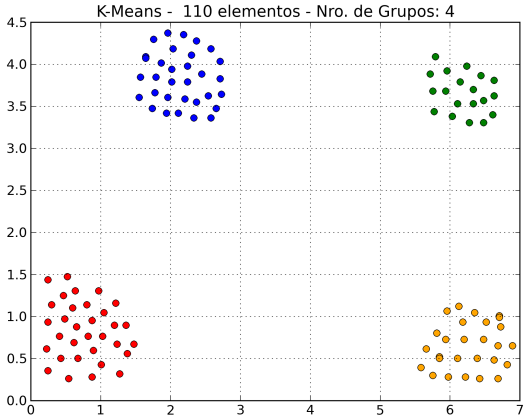
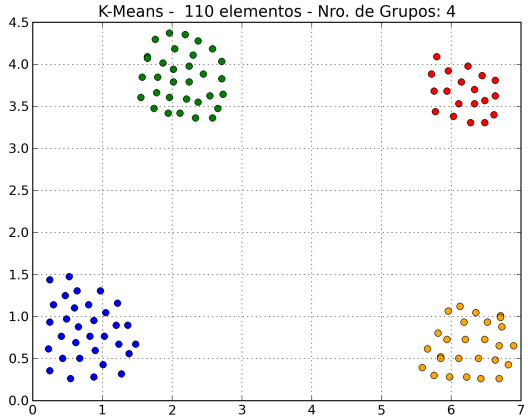


Figura 5.10 – Resultado obtido através do Q-SIM para a base de dados 2



(a) Primeira execução do algoritmo



(b) Segunda execução do algoritmo

Figura 5.11 – Resultado obtido através do *k-means* para a base de dados 2

Para o teste do DBSCAN, as entradas foram definidas com o mesmo valor do primeiro teste, 1 para o número mínimo de elementos e 0.6 para a distância mínima de similaridade. Os resultados obtidos no teste são apresentados na figura 5.12.

O DBSCAN também atingiu o objetivo de identificar os quatro grupos propostos pela base de dados 5.1b. Os resultados iguais para os algoritmos Q-SIM, DBSCAN e *k-means* demonstram que esse tipo de base de dados facilita a identificação dos padrões existentes nas informações. Um bom resultado para os algoritmos é que todos mantiveram a qualidade da similaridade das informações solicitada no teste com o Q-SIM, similaridade de 60%.

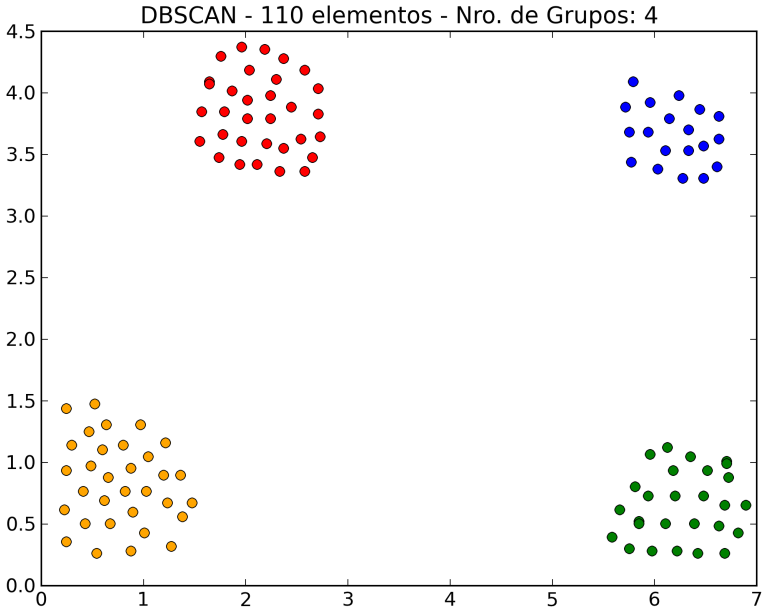


Figura 5.12 – Resultado obtido através do DBSCAN para a base de dados 2

O último algoritmo para teste foi o *Affinity Propagation*, que não necessita de nenhum parâmetro de entrada para determinar a quantidade de grupos e nem a similaridade entre os dados. O resultado do *Affinity Propagation* é demonstrado na figura 5.13.

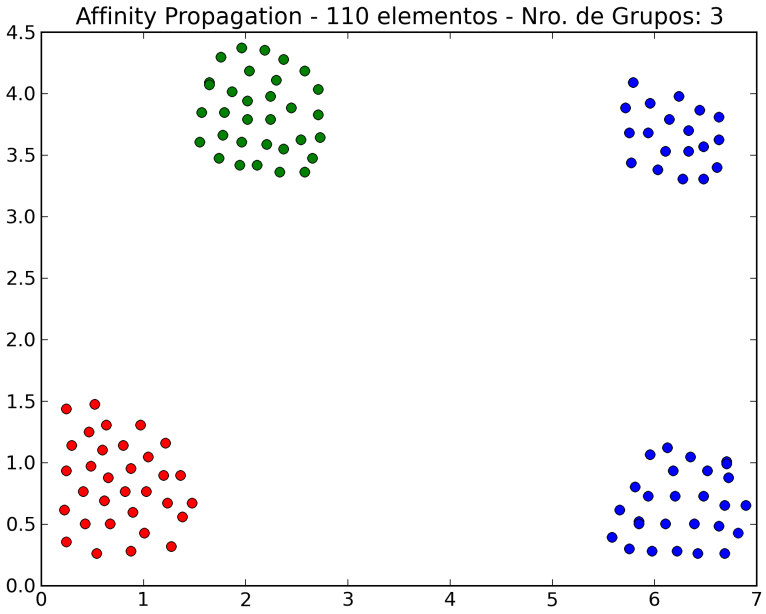


Figura 5.13 – Resultado obtido através do *Affinity Propagation* para a base de dados 2

O resultado obtido através do algoritmo *Affinity Propagation* não foi exatamente como o esperado. Ele gerou apenas três grupos para os quatro apresentados na base de dados 5.1b. Como esse algoritmo procura encontrar um ponto para servir de referência na criação do grupo,

como se fosse uma centroide, o *Affinity Propagation* encontrou um ponto entre os dois grupos existentes a direita que foi capaz de uni-los em um único grupo, vide figura 5.13. Contudo, não se pode afirmar que essa resposta, não é um possível agrupamento da base de dados, pode não ser a melhor resposta devido a quantidade de grupos esperados e também não mantém a similaridade de 60% entre os elementos do grupo como ocorreu com os outros três algoritmos.

Ao avaliar os índices para essa base de dados observou-se o mesmo resultado obtido durante a análise visual dos algoritmos. Os resultados foram aproximadamente os mesmos entre os algoritmos Q-SIM, DBSCAN e *k-means*, e com os piores índices está o *Affinity Propagation*. Na figura 5.14 é apresentado os valores encontrados para o índice de Dunn.

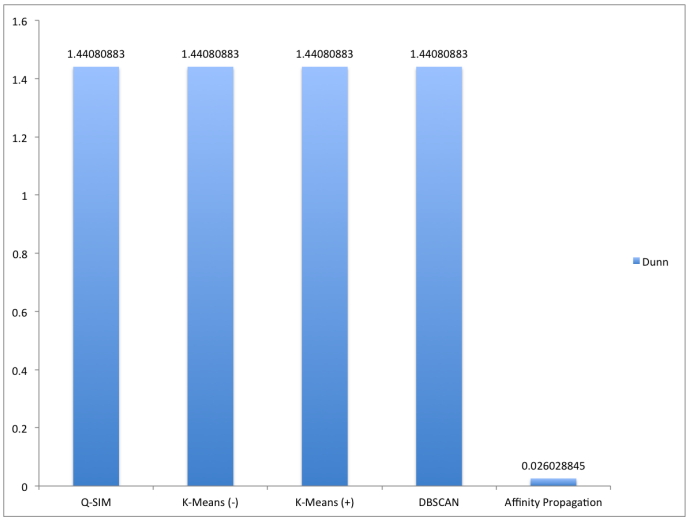


Figura 5.14 – *Dunn Index* para a base de validação 2

Os valores foram idênticos para todos os algoritmos exceto para o *Affinity Propagation* que obteve o pior índice, o que demonstra que o resultado com três grupos obtido pelo algoritmo diminui em aproximadamente 38% a similaridade entre os elementos dos grupos. O segundo índice analisado é o índice de Davies-Bouldin, que é demonstrado na figura 5.15.

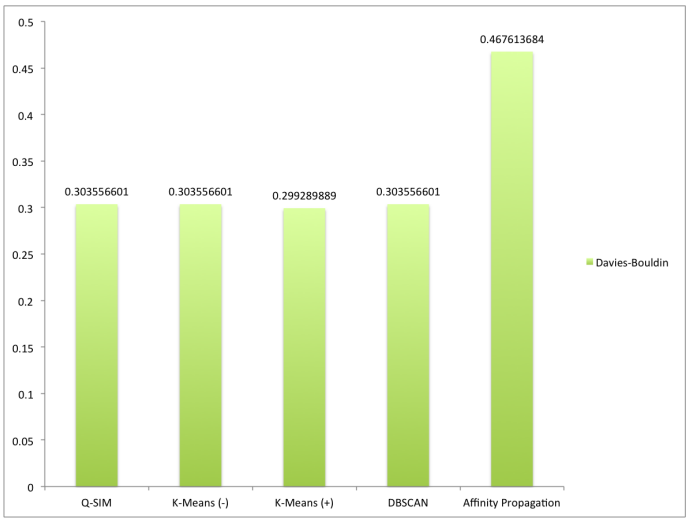


Figura 5.15 – *Davies-Bouldin Index* para a base de validação 2

Apesar de uma pequena oscilação no melhor resultado do *k-means*, ele, o Q-SIM e o DBSCAN apresentaram o mesmo resultado na média e deixando o *Affinity Propagation*, novamente, com o pior resultado. A diferença entre o índice do *Affinity Propagation* e os demais algoritmos é de 45%, o que o torna um resultado ruim para grupos que possuem pelo menos 60% de similaridade. Para finalizar as discussões da base de dados 5.1b, a variância dos grupos é exibida na figura 5.16.

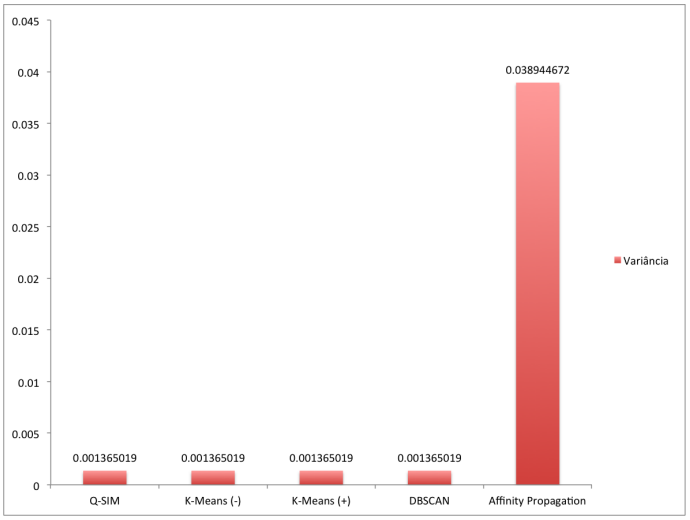


Figura 5.16 – Variância para a base de validação 2

Nesse índice o resultado foi conforme o esperado. Os algoritmos Q-SIM, DBSCAN e *k-means* apresentaram a melhor compactação, já que geraram quatro grupos. O pior nesse índice foi *Affinity Propagation* que devido ao grupo azul, apresentado na figura 5.13, obteve uma maior variância em seu resultado. Com exceção do *Affinity Propagation*, todos os algoritmos demonstraram ser melhores em pelo menos duas das três métricas para análise dos resultados, para classificação de grupos com pelo menos 60% de similaridade entre os seus elementos. A consolidação dos resultados é apresentada na tabela 5.2

Tabela 5.2 – Resultados dos Algoritmos Vs. Análises para Base de Dados 5.1b

Algoritmo	Analítico	Variância	Dunn	Davies-Bouldin
Q-SIM	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆
k-means	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆
DBSCAN	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Affinity Propagation	⋆	⋆	⋆	⋆

De acordo com o resultado consolidado na 5.2, foram 3 os melhores algoritmos para a base de dados 5.1b. Contudo, a escolha de melhores algoritmos é do Q-SIM e o DBSCAN, que encontraram os grupos de maneira automática, utilizando como base a similaridade entre os elementos e a distância entre os dados. Para esses dois algoritmos não foi necessário informar a quantidade de grupos desejada, como no *k-means*.

A terceira base utilizada na validação do Q-SIM, figura 5.1c, é composta de 145 pontos ao todo. Ela foi criada com a intenção de possuir um grupo grande e denso de pontos e um segundo grupo mais afastado e menor que o primeiro para identificar o comportamento dos algoritmos. A mesma quantidade de similaridade dos últimos dois teste foi solicitada, 60%. A figura 5.17 exibe o resultado encontrado pelo Q-SIM para a base de dados de número três.

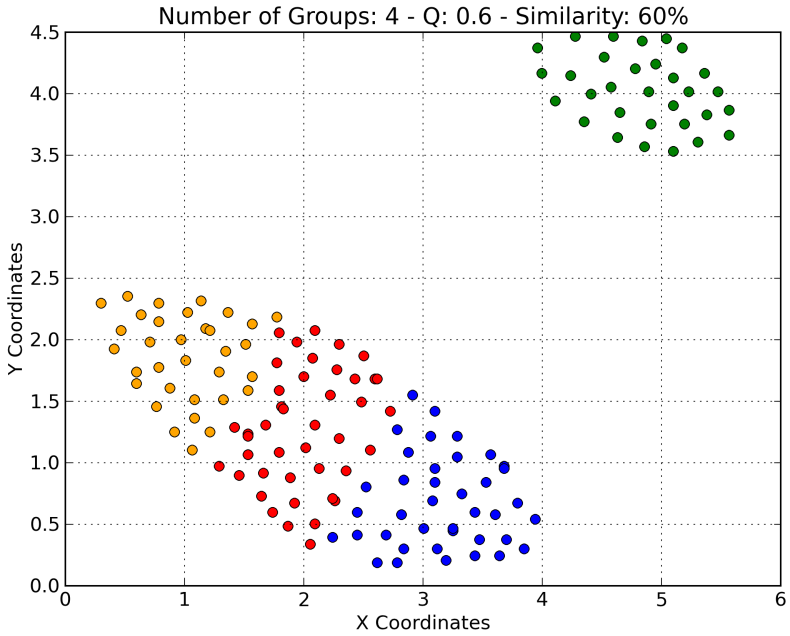


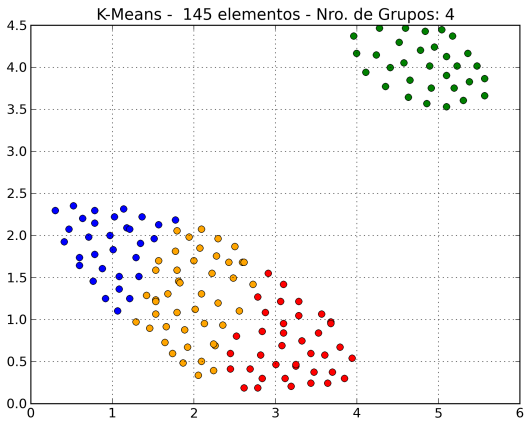
Figura 5.17 – Resultado obtido através do Q-SIM para a base de dados 3

O Q-SIM encontrou 4 grupos para essa base de dados. O grupo menor foi determinado como um dos quatro grupo encontrados, já o grupo maior foi necessário determinar três grupos para que a similaridade de 60% fosse mantida. Isso ocorreu devido ao fato do grupo maior, além de denso, possuir um diâmetro grande em suas pontas mais extremas de sua forma oval. Essa característica fez com que sua divisão fosse necessária para atender ao quesito da similaridade entre os elementos do grupo.

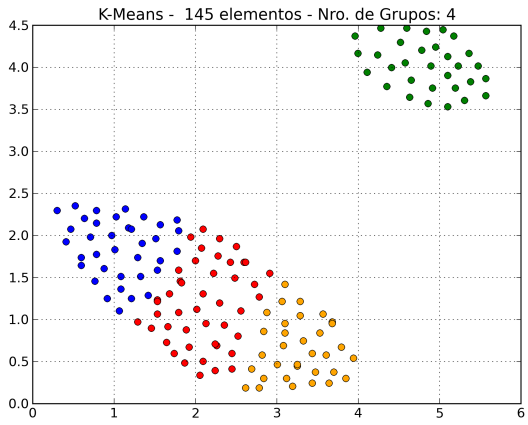
Com esse resultado em mãos, foi inserido como entrada do k-*means* o valor de quatro grupos desejados. O resultado é demonstrado na figura 5.18 na sequência da dissertação.

O comportamento obtido nos testes do k-*means* foram parecidos com o Q-SIM, no entanto sua aleatoriedade gerou diversos tamanhos entre o grupo maior, localizado na parte de baixo das figuras 5.18a e 5.18b, como pode ser observado. Essa diversidade na quantidade de elementos dentro dos grupos pode gerar em alguma das execuções, grupos que não possuam o grau de similaridade buscado durante os testes realizados. Isso torna o Q-SIM mais confiável para grupo qualitativamente similares.

O próximo algoritmo testado é o DBSCAN, que tem seu resultado exibido através da figura 5.19. Os parâmetros de entrada inseridos foram mantidos idênticos ao dos demais testes executados anteriormente.



(a) Primeira execução do algoritmo



(b) Segunda execução do algoritmo

Figura 5.18 – Resultado obtido através do *k-means* para a base de dados 3

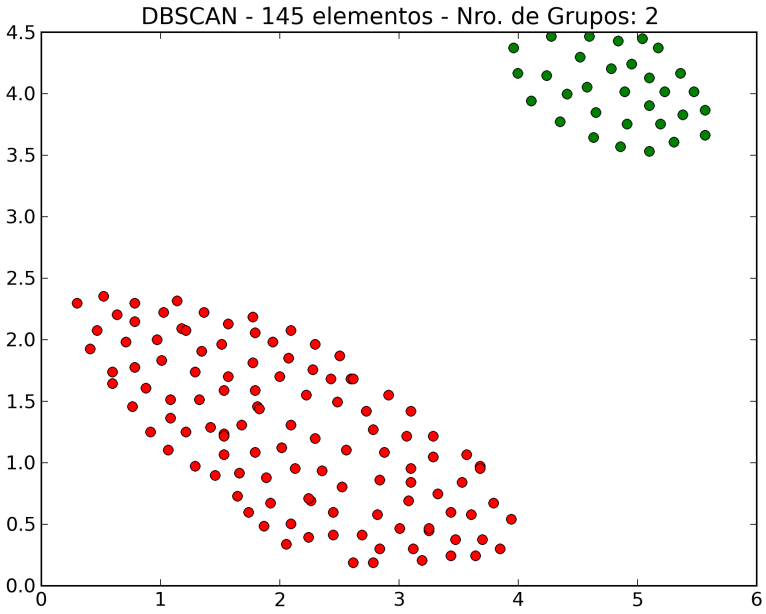


Figura 5.19 – Resultado obtido através do DBSCAN para a base de dados 3

O DBSCAN apresenta como resultado dois grupos no total. Apesar de existirem aplicações com essa necessidade, como por exemplo mapeamento geológico, a modelagem de usuário pode ser prejudicada com esse tipo de comportamento, pois pode generalizar os modelos que serão a base da construção da interface de um determinado sistema, por exemplo. Além do mais, a similaridade alvo dos testes não foi mantida para o grupo vermelho, devido aos pontos extremos do grupo serem pouco similares entre si. Nesse caso, o algoritmo Q-SIM também tem vantagem uma vez que ele procura manter a qualidade na similaridade entre os elementos dos grupos.

O quarto e último algoritmo a ser testado é o *Affinity Propagation*. Como não há a necessidade de informar valores para parâmetros de entrada do algoritmo, ele foi executado com base nas informações da base de dados 5.1c. O resultado apresentado está na figura 5.20.

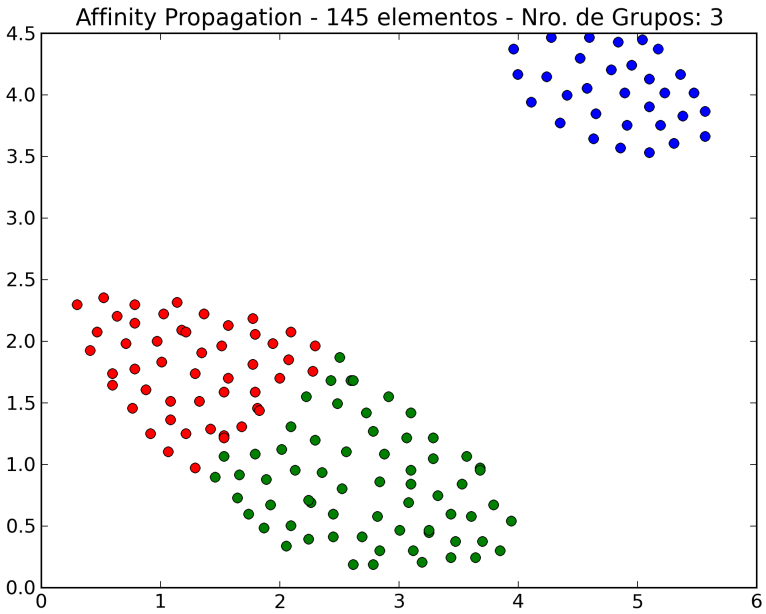


Figura 5.20 – Resultado obtido através do *Affinity Propagation* para a base de dados 3

Foram encontrados três grupos pelo algoritmo *Affinity Propagation*. O algoritmo manteve seu comportamento ao tentar agrupar o maior número de elementos em apenas um grupo. Esse comportamento fez com que os grupos formados a partir do grupo com maior número de pontos não ficasse homogêneo e causa-se um problema nas similaridades entre os elementos, que ficaram acima do solicitado no início dos testes que era de 60% pelo menos. Talvez para uma similaridade um pouco menor esse algoritmo atingiria o objetivo, porém para a similaridade de 60% o Q-SIM apresentou os melhores resultados analiticamente.

No entanto apenas a análise visual dos grupos formados não são totalmente válidas, pois é uma avaliação subjetiva aos resultados. Para resolver esse tipo de problema são utilizados alguns índices, já vistos anteriormente nesse capítulo, e o primeiro desses índices é o de Dunn. Os valores dos índices estão exibidos na figura 5.21.

Como pode-se observar na figura 5.21, o melhor índice entre os algoritmos foi do DBSCAN. Isso ocorre por que o índice de Dunn não possui um corte para uma similaridade máxima ou mínima solicitada. A comparação é feita pela compactação de um grupo e o quão distante um grupo esta do outro. Na sequência os melhores algoritmos foram o *k-means* e o Q-SIM, que possuem uma melhor compactação nos grupos, e por último ficou o *Affinity Propagation*.

O próximo índice apresentado é o índice de Davies-Bouldin. Esse índice possui o mesmo objeto do índice de Dunn, e consequentemente realiza o mesmo tipo de análise com

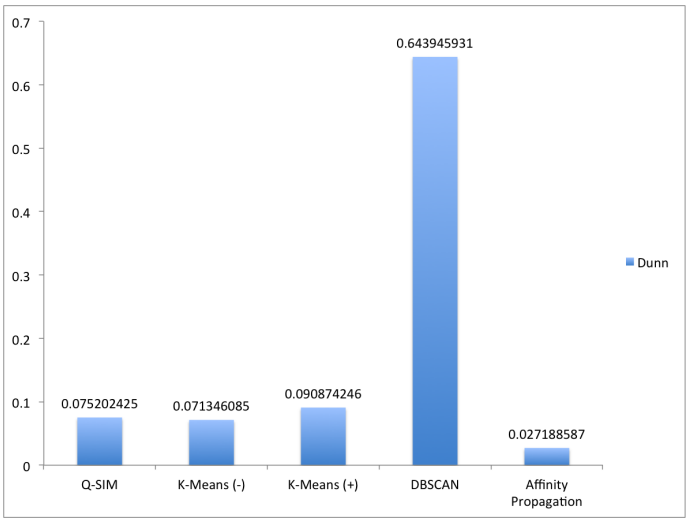


Figura 5.21 – *Dunn Index* para a base de validação 3

base nos grupos formados. A figura 5.22 exibe os resultados obtidos para o índice de Davies-Bouldin.

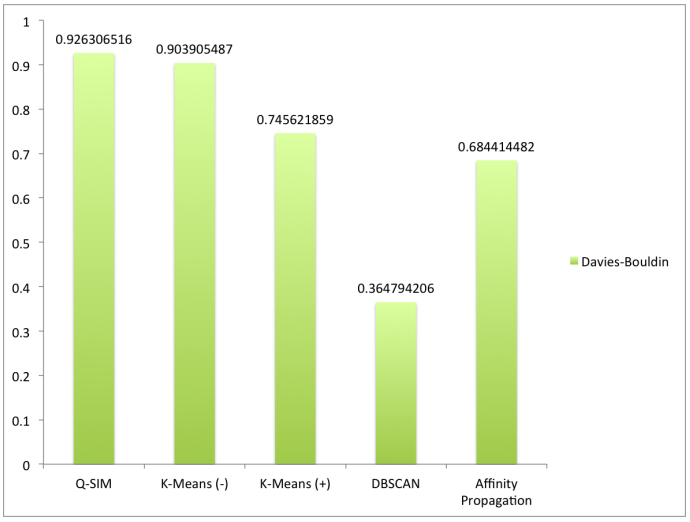


Figura 5.22 – *Davies-Bouldin Index* para a base de validação 3

Conforme esperado, devido ao resultado do índice de Dunn, o DBSCAN foi o que obteve o melhor resultado entre os algoritmos avaliados. O segundo melhor algoritmo foi o *Affinity Propagation*, devido ao número de grupos menor que os outros dois algoritmos. Ele é seguido pelo k*-means* que mesmo em seu pior resultado ainda foi melhor que o Q-SIM.

O último índice para a avaliação dos grupos formados é a variância dos grupos, que mede o quão compacto foram os grupos criados. Quanto maior a compactação dos grupos mais similares são elementos por ele contidos. O resultado da variância é apresentado na figura 5.23.

A compactação apresentada pelo k*-means* em sua média foi melhor a do Q-SIM, fazendo com que eles sejam os melhores algoritmos nesse quesito. O segundo melhor foi o DBSCAN, seguido na última posição da classificação pelo *Affinity Propagation*. Essa inversão ocorreu em relação ao dois primeiros índices, pois os grupos formados por eles são bem maiores dos que

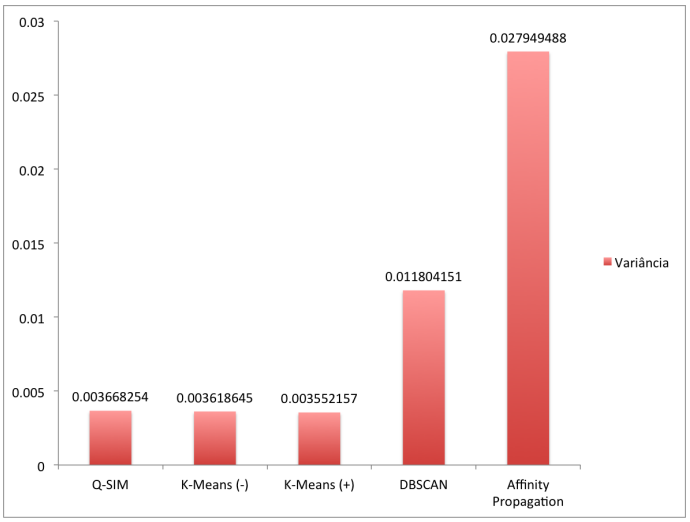


Figura 5.23 – Variância para a base de validação 3

os formados pelo Q-SIM e *k-means*. Devido a isso, as similaridades dos elementos dentro do grupo tendem a ser menores que os demais. A consolidação dos resultados é apresentada na tabela 5.3

Tabela 5.3 – Resultados dos Algoritmos Vs. Análises para Base de Dados 5.1c

Algoritmo	Analítico	Variância	Dunn	Davies-Bouldin
Q-SIM	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆	⋆
k-means	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆	⋆⋆
DBSCAN	⋆⋆⋆	⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆
Affinity Propagation	⋆⋆	⋆	⋆	⋆⋆⋆

Apesar dos resultados parecidos entre os algoritmos *k-means* e o Q-SIM, para a base de dados 5.1c, o melhor resultado foi obtido pelo *k-means*. Contudo, o resultado do *k-means* só foi melhor por que o número de grupos foi utilizado o mesmo gerado pelo Q-SIM. Caso contrário, o Q-SIM seria o algoritmo que melhor agruparia os dados com a similaridade de 60%.

A última base de dados utilizada para a validação do algoritmo Q-SIM e comparação com os demais está representada na figura 5.1d e possui um total de 170 pontos. Essa base de dados é composta de uma massa de dados densa que ao visualiza-lá aparenta apenas um grupo, porém dependendo do problema e do tipo de solução buscada deve-se ser divida em diversos grupos similares para não deixar a solução muito generalizada. Um problema onde determinar apenas um grupo para a base 5.1d pode ser negativo é o problema de modelagem de usuário.

Como o objetivo desse trabalho é a modelagem de usuários de um determinado sistema, executamos o Q-SIM solicitando os 60% de similaridade entre os elementos do grupo de acordo com os testes anteriores com o intuito de obter uma maior similaridade entre os perfis dentro dos grupos. O resultado desse teste é demonstrado através da figura 5.24.

Foram encontrados oito grupos com similaridade de pelo menos 60% entre os elementos do grupos existentes. Os grupos possuem um formato homogêneo entre si, mantendo o formato

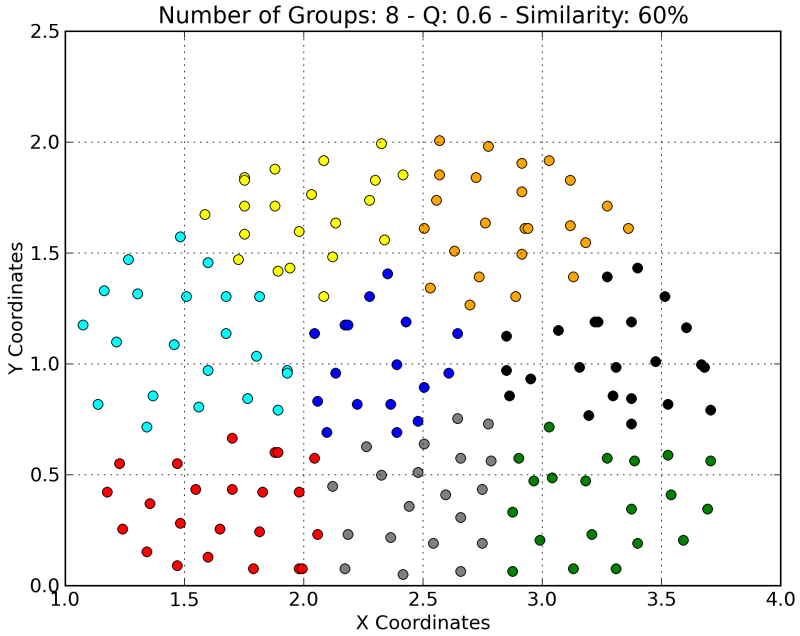
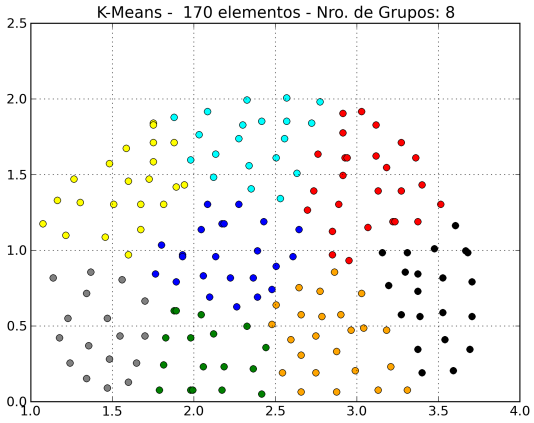


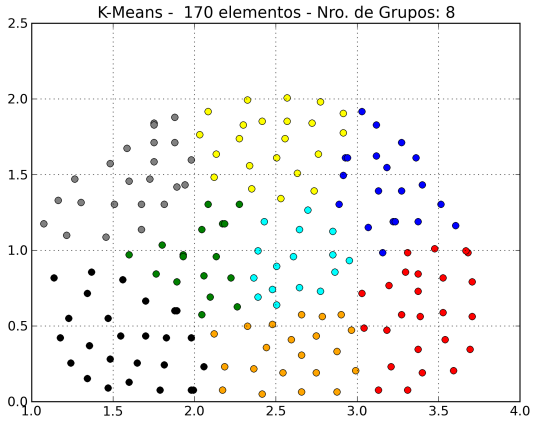
Figura 5.24 – Resultado obtido através do Q-SIM para a base de dados 4

e a quantidade de elementos parecidos. Isso pode ser considerado um bom resultado durante a modelagem de usuário, por não gerar grupos com uma representatividade maior que o outro, contudo esse fator depende mais do grau de similaridade e das informações utilizadas do que propriamente do comportamento do algoritmo. Esse resultado do Q-SIM é importante para garantir a melhor compactação dos grupos encontrados.

O segundo algoritmo utilizado foi o *k-means*, que será utilizado na comparação com o resultado do Q-SIM. A quantidade de grupos utilizados nesse teste foi de oito, ou seja, a mesma quantidade encontrada pelo Q-SIM no teste apresentado na figura 5.24. O resultado do *k-means* é exibido na figura 5.25.



(a) Primeira execução do algoritmo



(b) Segunda execução do algoritmo

Figura 5.25 – Resultado obtido através do *k-means* para a base de dados 4

Os grupos obtidos através do *k-means* possuem um formato parecido com os gerados pelo Q-SIM. Contudo, a aleatoriedade do *k-means* pode gerar alguns grupos maiores fazendo com que a qualidade na similaridade entre os elementos do grupo seja prejudica em sua análise. Os formatos obtidos nos testes foram bons já que os limites entre os grupos estão suaves e sem nenhum elemento causando alguma ruptura nesses limites.

A próxima análise realizada é sobre o algoritmo DBSCAN, que recebeu como parâmetro a distância mínima de 0.6, correspondente ao valor de 60% de similaridade, e o mínimo para se formar um grupo de 1 elemento. O resultado obtido no teste é apresentado na figura 5.26.

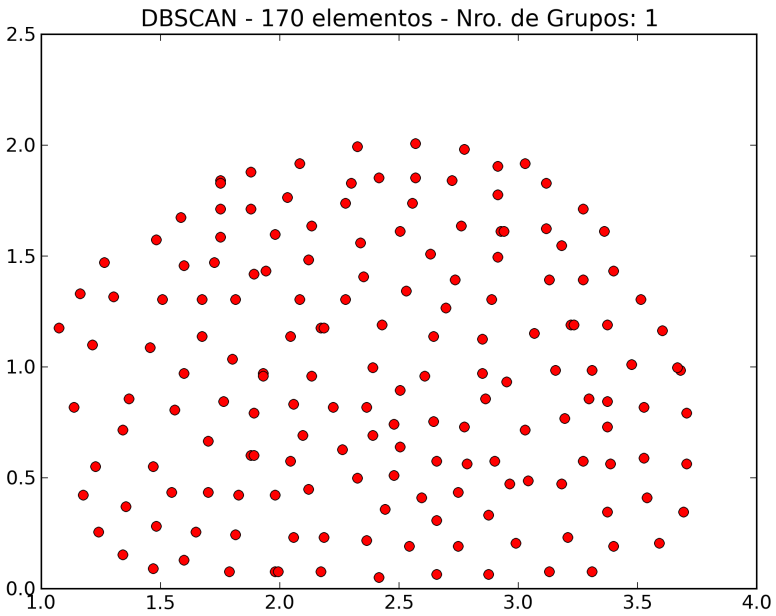


Figura 5.26 – Resultado obtido através do DBSCAN para a base de dados 4

O DBSCAN apresentou em seu resultado apenas um grupo para toda a massa de dados. Como já discutido anteriormente esse não é um resultado ruim dependendo da aplicação, porém ao tratar-se de modelagem de usuário esse tipo de resultado pode levar a uma generalização dos modelos prejudicando a segmentação de dados para tomada de decisão no projeto de interface.

O quarto e último algoritmo a realizar o teste é o *Affinity Propagation*, que seguindo os testes anteriores não é necessário informar nenhum parâmetro de similaridade ou quantidade de grupos para poder identifica-los. A figura 5.27 apresenta o resultado deste.

Quatro grupos foram formados pelo *Affinity Propagation* na base de dados 5.1d. Como parte de sua característica, um grande grupo azul escuro foi formado a esquerda da massa de dados em teste. Os demais grupos se formaram ao longo do processo procurando sempre agrupar o maior número de elementos em um único grupo. O comportamento do *Affinity Propagation* fez com que os formatos dos grupos não ficassem uniformes, e a qualidade da similaridade também não é atendida pelo algoritmo.

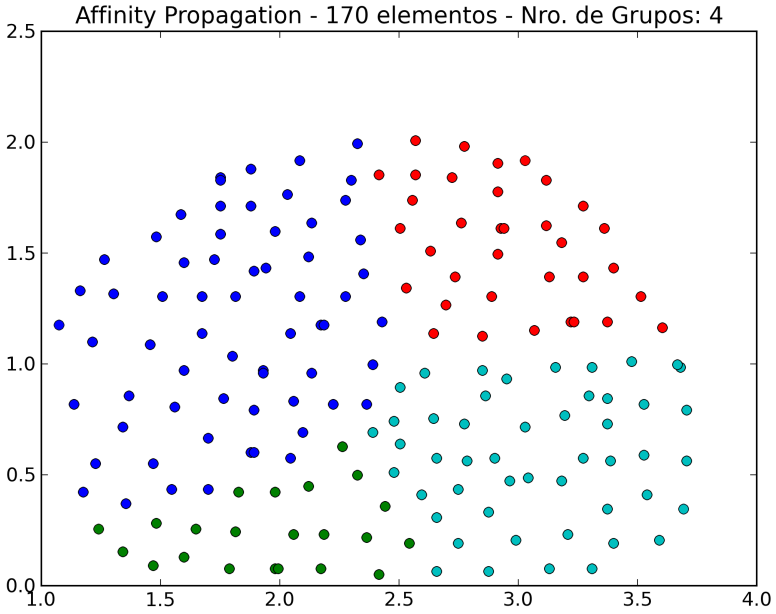


Figura 5.27 – Resultado obtido através do *Affinity Propagation* para a base de dados 4

O resultado no entanto pode ser analisado, em modelagem de usuário, como os grupos mais representativos sendo maiores e o menos representativos sendo os grupos menores. Porém, esse tipo de afirmação só poderá ser realizada analisando também a regra de similaridade utilizada para análise e aquisição dos padrões dos dados em estudo. Outro ponto importante é que a qualidade na criação dos grupos é perdida quando obtêm-se grupos muito grandes e necessita de uma grande similaridade entre os perfis dos usuários. Tais informações devem ser analisadas ao fim de cada um dos projetos de interface que utilizará o modelo gerado pelo algoritmo.

Assim como realizado para os outros testes, os índices de avaliação dos grupos são apresentados e discutidos na sequência. O primeiro índice a ser apresentado é o índice de Dunn, que valida a diferença entre os elementos de grupos distintos e a semelhança entre os elementos do mesmo grupo. A figura 5.28 exibe os resultados do índice de Dunn.

Apesar do k-*means* em seu melhor caso apresentar o melhor resultado para o índice, o Q-SIM quando comparado com a média entre os dez resultados do k-*means* (0.092) é o melhor algoritmo para o índice de Dunn devido a sua constância nos grupos gerados. O DBSCAN apresentou um índice muito pequeno que levou a ser o pior dos algoritmos para o índice. Entretanto, esse índice leva em consideração a criação de dois ou mais grupos para o seu cálculo e como o DBSCAN gerou apenas um grupo não é possível avaliar ele por esse método.

O resultado do *Affinity Propagation*, apesar de ter um menor número de grupos, apresentou o pior índice dos três algoritmos que conseguiram valores para este. A similaridade entre os elementos dos grupos foi baixa e entre elementos extra grupos não foi baixa como espera-se, isso fez com que o resultado do *Affinity Propagation* fosse classificado como o pior algoritmo para a base de dados 5.1d, segundo o índice de Dunn.

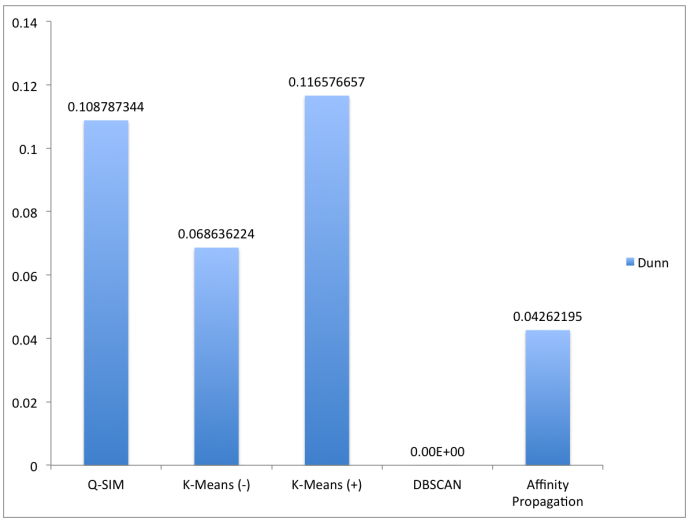


Figura 5.28 – *Dunn Index* para a base de validação 4

O segundo índice utilizado na avaliação dos resultados é o índice de Davies-Bouldin, que possui o mesmo objetivo do índice de Dunn, mas nesse índice o melhor resultado é o menor índice. Os índices de Davies-Bouldin obtido por cada um dos algoritmos em análise é exibido através da figura 5.29.

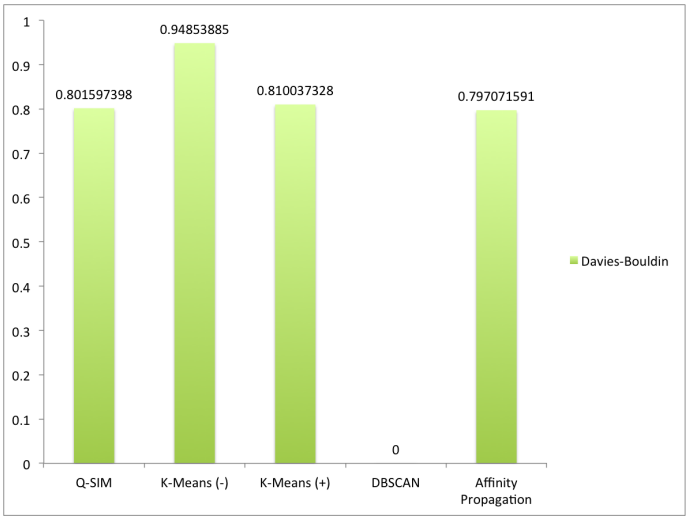


Figura 5.29 – *Davies-Bouldin Index* para a base de validação 4

No índice de Davies-Bouldin o melhor resultado apresentado é o do algoritmo *Affinity Propagation*, que ficou a frente do Q-SIM por apenas alguns milésimos, seguido pelo k-*means* que com a média de 0.86 é o pior algoritmo para essa base de dados. O DBSCAN também não obteve o índice de Davies-Bouldin devido a quantidade de grupos gerados ser igual a 1. Como o índice de Dunn, o índice de Davies-Bouldin também necessita de dois ou mais grupos para o seu cálculo.

Por fim, o índice para avaliar os resultados dos algoritmos é a variância com o objetivo de medir a compactação e a similaridades intra grupos. Na figura 5.30 são exibidos os valores para cada um dos algoritmos.

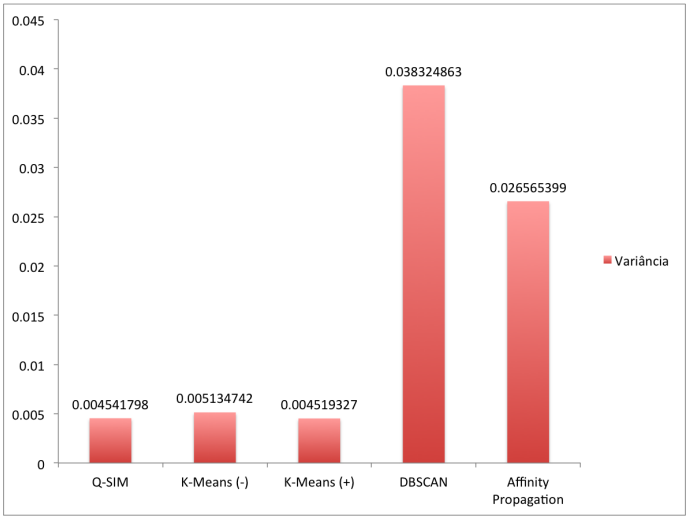


Figura 5.30 – Variância para a base de validação 4

A pior variância entre os algoritmos é do DBSCAN que ficou muito maior do que a dos outros algoritmos, justamente devido único grupo gerado por ele. Para esse índice, foi possível obter um resultado do *Affinity Propagation* que ficou com a segunda pior classificação entre os quatro algoritmos, originado pela alta variação dos dados entre os quatro grupos gerados.

O melhor resultado pelas variâncias foi dado ao melhor caso do k-*means*. Entretanto, a média da variância dos dados entre os resultados do k-*means* é de 0.0048. Esse resultado aponta novamente o Q-SIM como o melhor algoritmo no índice da variância dos dados, tornando-o o algoritmo que melhor apresenta similaridade entre elementos intra grupos. A consolidação dos resultados é apresentada na tabela 5.4

Tabela 5.4 – Resultados dos Algoritmos Vs. Análises para Base de Dados 5.1d

Algoritmo	Analítico	Variância	Dunn	Davies-Bouldin
Q-SIM	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆
k-*means*	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆
DBSCAN	⋆	⋆		
Affinity Propagation	⋆⋆	⋆⋆	⋆⋆	⋆⋆⋆⋆⋆

De acordo com os resultados consolidados na tabela 5.4, o algoritmo com o melhor resultado para a base de dados 5.1d, é o Q-SIM. O Q-SIM apresentou o melhor similaridades entre os elementos dos grupos obtidos, com o menor número possível de grupos. Nesse resultado, não foi possível fazer a comparação com o DBSCAN nos índices de Dunn e Davies-Bouldin, pois este gerou apenas um grupo e esses índices necessitam de pelo menos dois grupos para que seja possível o cálculo dos índices.

Os resultados apresentados ao longo dos testes com as quatro base de dados, demonstra que cada um dos algoritmos possui melhores resultados em algum dos índices e analises das bases. Contudo, o único algoritmo com característica de manter o grau de similaridade entre

os elementos com um determinado valor mínimo é o Q-SIM, que utiliza esse parâmetro para criação dos grupos.

O k-*means* apesar de apresentar essa característica, como o Q-SIM, só fez isso pois a quantidade de grupos gerados por ele foi a mesma que o Q-SIM encontrou. Deve-se lembrar que quando o k-*means* é executado a informação de quantos grupos deve ser informada pelo especialista, que analisa os índices apresentados e algumas outras informações para confirmar se a quantidade de grupos é suficiente para o projeto.

Outro fato importante para acrescentar nesta discussão é que os melhores índices do k-*means*, nunca foram resultantes de um único agrupamento gerado como os demais algoritmos. Todos os melhores valores para os índices do k-*means* ocorreram separadamente uns dos outros, sendo a única exceção a base de dados 5.1b que possui os quatro grupos bem definidos.

Quanto aos resultados apresentados pelo DBSCAN e *Affinity Propagation*, são bons resultados conforme a aplicação que está em análise, porém para o problema de modelagem de usuário o Q-SIM é o algoritmo mais aconselhado. Isso ocorre, pois a generalização ou especialização dos resultados depende diretamente do especialista de acordo com a variação da similaridade. Entretanto, a qualidade da similaridade solicitada nunca será ferida na utilização do Q-SIM, diferentemente dos outros algoritmos.

Na sequência serão apresentados alguns resultados do Q-SIM com diferentes valores para Q nas bases de dados utilizadas durante a validação do algoritmo, que foi provada ao longo desse capítulo. Alterado o valor Q para 0.4, pode-se verificar os diferentes resultados para as bases 5.1a e 5.1b nas figuras 5.31 e 5.32, respectivamente.

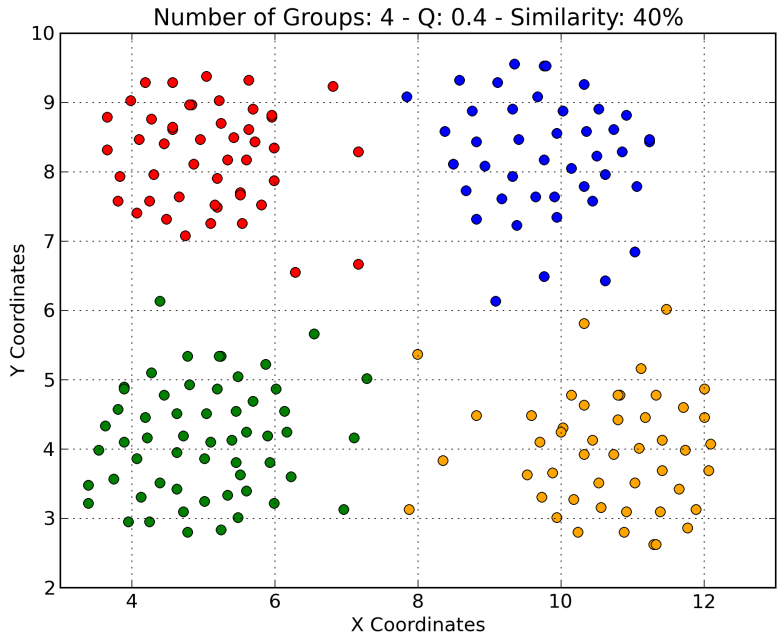


Figura 5.31 – Resultado obtido através do Q-SIM para a base de dados 1 com Q igual a 0.4

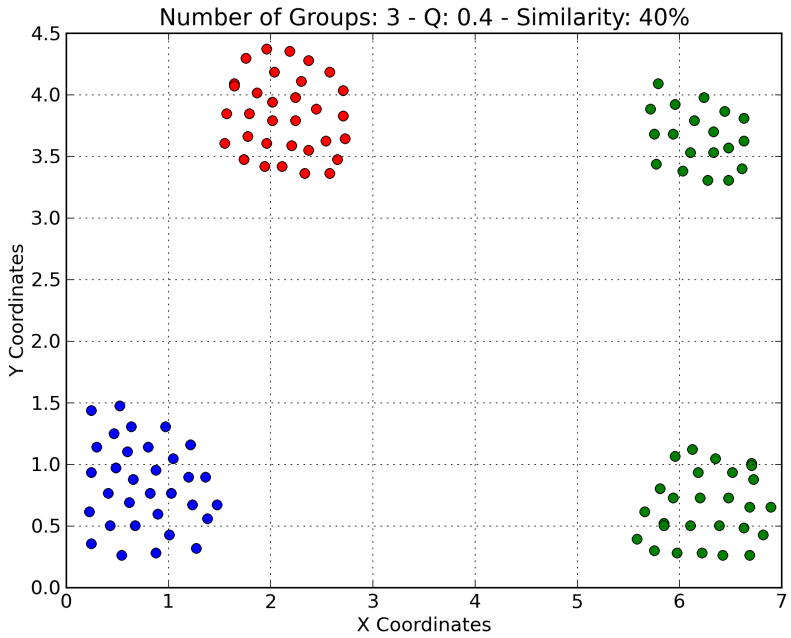


Figura 5.32 – Resultado obtido através do Q-SIM para a base de dados 2 com Q igual a 0.4

Na figura 5.31, o Q-SIM encontrou quatro grupos bem divididos, comportando-se bem mesmo com os ruídos existentes entre os grupos. Já o resultado da figura 5.32 demonstra o mesmo resultado obtido pelo *Affinity Propagation*, vide figura 5.13, para a base de dados 5.1b. Dessa forma, podemos observar que a similaridade entre os elementos no resultado do *Affinity Propagation* é mais baixa do que a procurada durante o teste, mas ainda é um resultado válido para uma similaridade de 40%.

A próxima variação a ser apresentada é a entrada do valor Q em 0.2, correspondendo a 20% de similaridade. As bases utilizadas para a demonstração foram as seguintes, 5.1c e 5.1d. Os resultados são exibidos nas figuras 5.33 e 5.34.

O resultado obtido no teste com a base 5.1c para o valor de Q em 0.2, foi o mesmo resultado obtido pelo DBSCAN para a mesma base de dados, vide figura 5.19. Isso demonstra que o Q-SIM pode atingir os mesmos resultados que alguns dos demais algoritmos, e melhorar os valores dos índices. Contudo, deve-se lembrar que o Q-SIM trabalha com a qualidade desejada pelo especialista nos elementos contidos nos grupos e as variações dos resultados irão depender única e exclusivamente do valor Q, não dependendo de analisar os índices para atribuir um melhor resultado.

A última variação a ser apresentada é na base de dados 5.1d, que obteve-se o resultado da figura 5.34 para o valor de Q em 0.2. Nesse teste encontrou-se três grandes grupos dentro da massa de dados de 170 pontos. Esse resultado foi um pouco melhor do que os quatro grupos obtidos pelo *Affinity Propagation* no resultado para essa base de dados, vide figura 5.27. O resultado foi melhor justamente pela compactação entre os grupos realizada pelo Q-SIM, ao in-

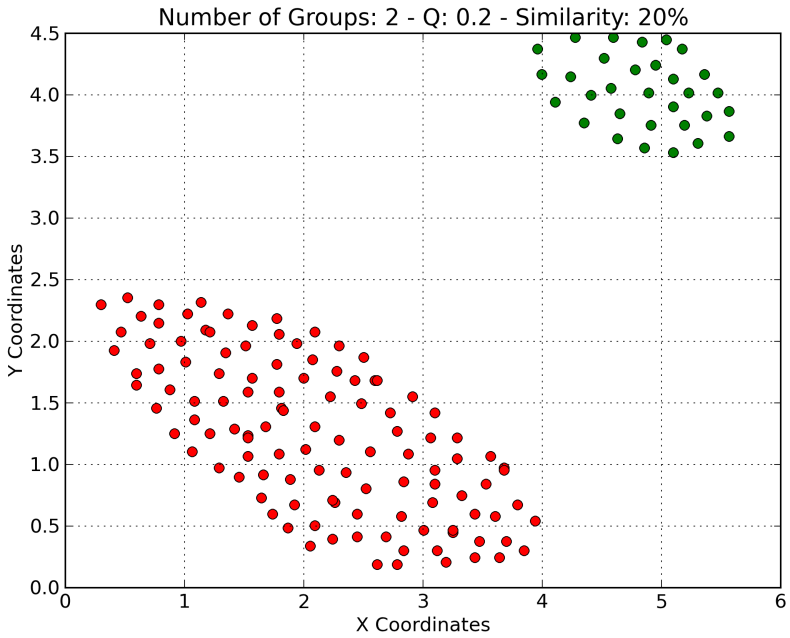


Figura 5.33 – Resultado obtido através do Q-SIM para a base de dados 3 com Q igual a 0.2

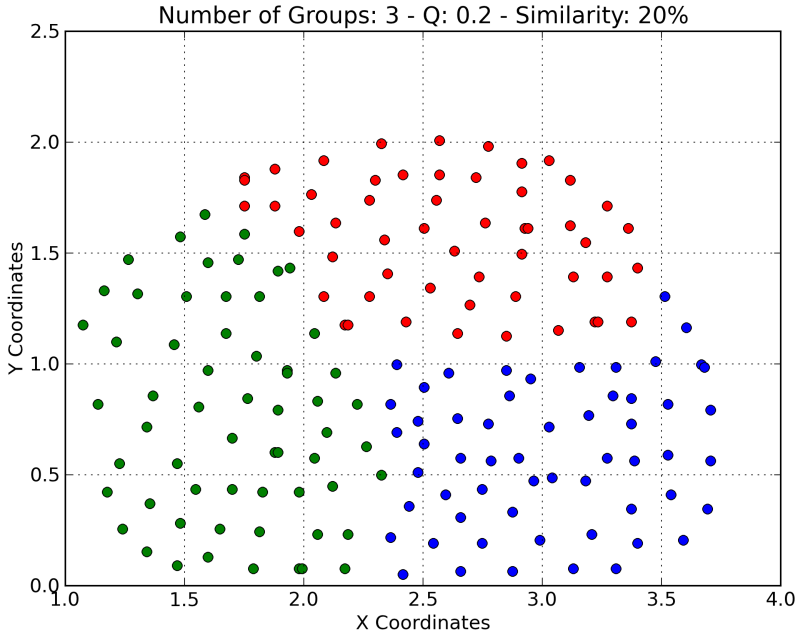


Figura 5.34 – Resultado obtido através do Q-SIM para a base de dados 4 com Q igual a 0.2

vés dos grupos não uniformes obtidos através do *Affinity Propagation* que geram uma variância muito alta, perdendo a similaridade entre os elementos intra grupo.

De acordo com os resultados obtidos durante a avaliação do Q-SIM, pode-se dizer que os índices de Dunn e Davies-Bouldin não expressão exatamente o melhor resultado para o agrupamento realizado. Esses índices sofrem influências diretas do número de grupos e a quantidade de elementos existentes nesses grupos. Os índices devem sempre ser analisados de acordo com

a problemática do cenário de agrupamento para que a tendência dos resultados seja minimizada ou até anulada. Baseado nessa análise o Q-SIM demonstrou-se o melhor algoritmo dentro de uma média para todas as bases de dados apresentadas durante a avaliação dos algoritmos.

Outro ponto a ser analisado em relação ao processo de *clustering* em situações que deseja-se acompanhar a evolução dos grupos ou reutilizar a classificação gerada para associar a algum processo dentro do sistema. Nesse tipo de situação é importante manter resultados estáveis para que possam ser utilizados ao longo do tempo sem mudanças indesejadas ou aleatoriedade nos resultados. Nesse ponto em especial, o k-*means* é o algoritmo menos aconselhável, pois seu processo é totalmente aleatório. Os outros 3 algoritmos, inclusive o Q-SIM, são totalmente estáveis em seus resultados, o que favorece em problemas que a reutilização constante dos grupos é necessária como por exemplo, modelagem de usuário para interface adaptativa.

Além disso, o algoritmo Q-SIM é capaz de gerar grupos a partir de um valor Q de qualidade de similaridade, mantendo a maior similaridade solicitada entre todos os membros de um grupo. Isso o torna o algoritmo mais aconselhado para o processo de criação de personas. Sendo assim, é aplicado o método proposto na seção 4 no projeto FINEP, o PEAD-PMPT, utilizando o Q-SIM como algoritmo de agrupamento.

6 APLICANDO O Q-SIM NO PROJETO PEAP-PMPT

Para realizar o processo de criação de Personas junto ao projeto PEAD-PMPT foi necessário o desenvolvimento de um componente para capturar as informações dos usuários durante a utilização do sistema. Esse componente foi escrito na linguagem de programação *javascript*, já que o sistema do projeto é baseado na *web*, utilizando o *framework* JQuery. As informações dos usuários coletadas são as sete variáveis apresentadas na seção 6.1.

Todas informações capturadas são armazenadas no banco de dados da aplicação identificando o usuário através da sessão de acesso. Com os dados armazenados, algumas *procedures* em PL/SQL foram criadas para extrair as informações de cada um dos usuários. Essas informações são salvas em um arquivo texto (para facilitar a comunicação entre os sistemas de linguagens diferentes), que é informado ao algoritmo Q-SIM, onde a partir destas é realizado o agrupamento dos usuários baseado em suas similaridades. A regra de similaridade para esse processo consiste em calcular a similaridade local utilizando a equação 4.5 e posteriormente efetuar o cálculo da similaridade global, representada através da equação 4.4.

Essa regra de similaridade foi escolhida, pois demonstrou a melhor opção para o problema de identificar as similaridades entre os perfis dos usuários (seção 4.1.1). Por fim, cada um dos grupos passa pelo processo de criação das Personas utilizando a regra da média, mediana e moda apresentada na seção 4. O processo de criação das Personas do projeto será detalhado na seção 6.2.

6.1 Capturando informações do usuário automaticamente

Ao longo dessa pesquisa, percebeu-se que grande parte dos estudos realizados para criação de modelos de usuários, inclusive Personas, utiliza informações providas por questionários ou entrevistas realizados junto aos potenciais usuários dos sistemas. Tais métodos, dependendo da situação do usuário, pode levar a captura de ruídos. Esses ruídos são informações divergentes da real situação dos usuários, causados pelos usuários que se sentem constrangidos quando questionados sobre uma situação especial àquele momento. Esse tipo de cenário pode levar a omissão da resposta real, prejudicando a modelagem e análise do projeto de interface direcionada à necessidade do usuário devido ao ruído na informação (MASIERO et al., 2011).

A solução para este problema é realizar a captura das informações de maneira automática e transparente para o usuário. Com o objetivo de atender essa demanda, optou-se por utilizar o método de captura de dados criado por D'Angelo (2012) onde é apresentado um componente integrado ao sistema que realiza a captura automática das informações de utilização do

usuário. Assim, é possível obter as informações do usuário, livre de ruídos e de tal forma, que as informações capturadas sejam natural a ação do usuário conforme o uso do sistema.

Durante a análise do trabalho apresentado por D'Angelo (2012) percebeu-se que foram mapeadas ao todo 28 variáveis representando a manipulação do sistema por parte usuário. As variáveis determinadas por D'Angelo (2012) são apresentadas a seguir:

1. Tempo de navegação total
2. Tempo de navegação em página de conteúdo
3. Tempo de navegação em página de questionário
4. Tempo médio para preenchimento dos campos texto de todos os formulários
5. Tempo gasto no preenchimento dos campos não textuais
6. Tempo médio para preenchimento dos campos textos intuitivos (ex.: Nome)
7. Quantidade de toques por segundo em todos os campos textos
8. Quantidade de toques por segundo em todos os campos textos intuitivos (ex.: Nome)
9. Porcentagem de uso da tecla *backspace* para os campos textuais
10. Porcentagem de uso da tecla *backspace* para os campos textuais intuitivos (ex.: Nome)
11. Ocorrência de erros em preenchimentos dos formulários
12. Quantidade de vezes que ocorreu erro no preenchimento do mesmo formulário (duas ou mais vezes)
13. Quantidade de ocorrências de duplos cliques em um mesmo link
14. Quantidade de vezes do acionamento do botão ajuda
15. Quantidade de vezes que o usuário utilizou o menu
16. Utilização dos links do tipo *breadcrumbs*
17. Quantidade de vezes o usuário utilizou a busca
18. Quantidade de vezes o usuário utilizou o *auto-complete* na busca
19. Quantidade de vezes que o usuário clicou em links ou botões dos destaques da página principal do site
20. Tempo de movimentação do mouse na área de conteúdo do site
21. Tempo de movimentação do mouse na área de menu do site
22. Tempo de movimentação do mouse na área de busca do site

23. Quantidade de cliques incorretos
24. Quantidade de vezes que o usuário acessou uma página de conteúdo detalhado
25. Acesso a um mesmo conteúdo duas ou mais vezes
26. Tempo médio de visualização das páginas de conteúdo
27. Tempo total de visualização das páginas informativas como política de privacidade
28. Tempo total de visualização da página inicial

Quando analisada as variáveis acima com mais detalhe percebeu-se que essas mapeavam tarefas que são dependentes, não apenas do usuário, mas também da interface ou do cenário ao qual o usuário atua. Por exemplo a tarefa 16 (Utilização de links do tipo *breadcrumbs*), que é dependente da navegação que a interface possibilita. Sendo assim, realizou-se um estudo para filtrar as variáveis que fossem dependentes apenas das ações, comportamentos e habilidades do usuário, sem nenhum outro tipo de dependência direta.

Outro ponto considerado neste estudo foi o fato de que apesar de utilizar-se uma interface web para o componente, o sistema poderia ser acessado através diversos dispositivos, como computadores, *tablets* e *smartphones*. Dessa forma, foram selecionadas 7 variáveis das 28 apresentadas. Os casos das variáveis escolhidas 1 e 2, na sequência do texto, foi realizado um agrupamento das separações realizadas por D'Angelo (2012), procurando simplificar o método de captura durante a utilização, já que essas não demonstraram grande variação entre os valores coletados. A tabela 6.1 apresenta as análises realizadas para a escolha das 7 variáveis.

As sete variáveis escolhidas para compor a lista final de variáveis utilizadas nessa dissertação são:

1. Tempo de preenchimento dos campos na interface
2. Velocidade de digitação
3. Porcentagem do uso da tecla *backspace*
4. Quantidade de erros ao preencher um formulário
5. Quantidade de vezes que ocorreu 2 ou mais erros ao preencher um mesmo formulário
6. Quantidade de vezes que ocorreu um duplo clique em um link
7. Quantidade de cliques na interface fora de um botão ou link (Cliques incorretos)

Com as variáveis escolhidas, configura-se o componente para a captura das informações. Assim, conforme o usuário utiliza o sistema, as informações são coletadas. Posteriormente, depois de armazenar as informações, é possível realizar o trabalho de análise e manipulação dos dados para construir as Personas do sistema. O processo de criação das Personas será discutido em mais detalhes na próxima seção.

Tabela 6.1 – Análise feita para seleção das variáveis para captura do comportamento do usuário

Variáveis	Análise/Justificativa
1, 2 e 3	Foram excluídas da seleção, já que são variáveis dependentes da tarefa executada, isso dificulta a análise, pois há necessidade de mapear todos os cenários para cada sistema.
4, 5 e 6	Foram utilizadas para compor a variável 1 da lista das escolhidas, pois apresentaram uma pequena variação na análise dos resultados de D'Angelo (2012).
7 e 8	Foram utilizadas para compor a variável 2 da lista das escolhidas, pois apresentaram uma pequena variação na análise dos resultados de D'Angelo (2012).
9 e 10	Foram utilizadas para compor a variável 3 da lista das escolhidas, pois apresentaram uma pequena variação na análise dos resultados de D'Angelo (2012).
11, 12 e 13	Foram selecionadas sem nenhuma alteração, pois também são dependentes do usuário.
14 à 22	Foram excluídas, pois são variáveis dependentes da interface e do dispositivo que será utilizado.
23	Foi selecionada sem nenhuma alteração, pois também são dependentes do usuário.
24 à 28	Foram excluídas, pois as variáveis são dependentes da tarefa a qual o usuário realizará.

6.2 Resultados do Projeto PEAP-PMPT

Durante a fase de levantamento de requisitos do projeto foram entrevistados usuários com diferentes funções. Ao longo das entrevistas foram observados diferentes características entre cada um dos profissionais que fazem parte dos grupo de usuários do sistema. Essas características foram armazenadas em notas feitas pelo especialista para criação das Personas que guiam o processo de melhoria da interação do sistema.

Baseado nas informações observadas e armazenadas durante a fase de entrevistas, o especialista cria a primeira versão das Personas para o início do projeto. O texto em cor preta apresentado na Persona 6.3, por exemplo, foi definido pelo especialista nessa fase do processo. Este texto é o que apresenta maior dificuldade para modificar ou criar automaticamente, pois ele é subjetivo e depende da observação feita pelo especialista.

Com a fase de levantamento de requisitos concluída, o próximo passo é o desenvolvimento do sistema onde o componente de captura de informações do usuário é construído. A partir de uma versão funcional do sistema é possível a realização dos testes com o usuário e iniciar a coleta das informações para análise e melhoria do sistema.

Foram coletados informações de 154 usuários ao longo dos testes. Os usuários são formados pela equipe de testes funcionais do sistema, médicos do Hospital Heliópolis e alguns estudantes que auxiliaram nos testes de usabilidade. Com a base de dados formada, foram

realizadas alguns testes verificando os resultados das Personas obtidas. O valor de Q foi variado entre os valores 0.2, 0.4, 0.6 e 0.8, representando a similaridade desejada, e o conjunto de informações obtidas em cada um desses testes foi 1, 3, 3 e 5, respectivamente.

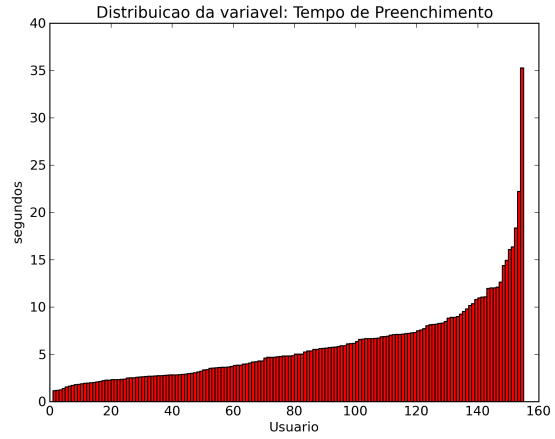
Os resultados que geraram 1 e 3 conjuntos foram generalizados. Dentre os 3 conjuntos gerados, um grupo conteve praticamente todos os perfis coletados e os demais grupos ficaram bem específicos contendo apenas 1 perfil para o caso de cada um dos dois grupos restantes. O resultado obtido como valor Q de 0.8, distribuiu melhor os perfis dentro dos grupos devido a maior similaridade solicitada. Os conjuntos de informações obtidos através do Q-SIM são apresentados na tabela 6.2, sendo que cada linha da tabela representa um conjunto de informações dos perfis do sistema, que auxiliará na criação das Personas. Para entender melhor a distribuição das variáveis capturas dos usuários e compreender as Personas geradas na tabela 6.2, a figura 6.1 apresenta os valores em forma de gráfico sendo que o eixo x representa os usuários e o eixo y o valor de cada uma das variáveis.

Tabela 6.2 – Informações obtidas através da coleta sistema PEAD-PMPT para auxiliar na geração dos Personas

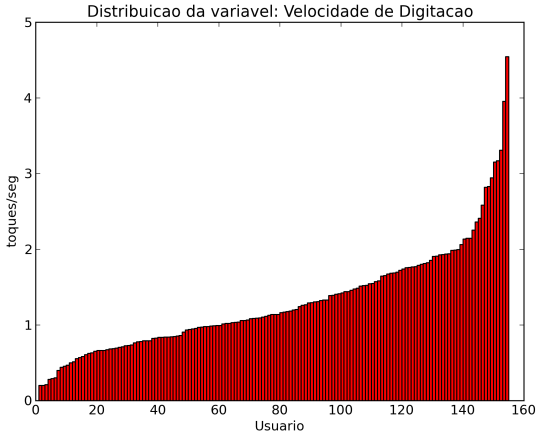
Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
1.21 toques/seg	4.17 seg	8.33%	16 erros	2 vezes	0
1.31 toques/seg	5.49 seg	1.47%	3 erros	1 vez	1 duplo clique
1.46 toques/seg	6.41 seg	48.52%	5 erros	1 vez	0
0.92 toques/seg	5.94 seg	75%	7 erros	1 vez	0
1.11 toques/seg	11.46 seg	23.22%	2 erros	1 vez	0

A variável de clique incorreto foi removida das Personas resultantes, pois a quantidade de elementos de interface existentes em uma tela, que podem sofrer a ação do clique e executar alguma tarefa, aumentou desde a pesquisa apresentada por D'Angelo (2012), onde essas eram consideradas como ações de cliques incorretos. Dessa maneira, utilizar essa variável para realizar a análise e construir uma persona se torna inviável para esse modelo. Contudo, essas informações podem ser utilizadas para mapear o comportamento do usuário em uma determinada tarefa e analisar qual é o melhor caminho para o sucesso da tarefa, por exemplo, ou até mesmo quais são os atalhos mais adequados para cada perfil. Como esse não é o objetivo dessa dissertação este estudo não foi realizado aqui e apenas excluiu-se a variável dos conjuntos bases para formação das Personas.

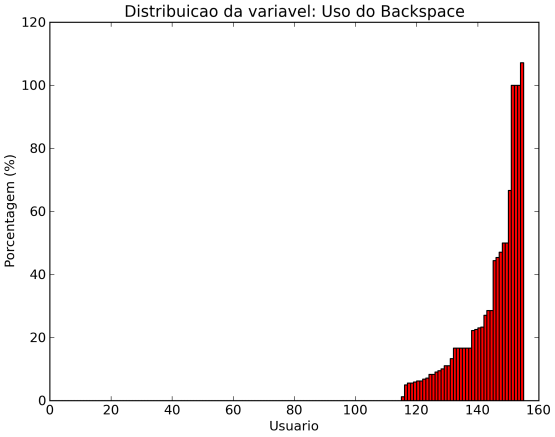
Com os conjuntos de informações gerados a partir dos grupos obtidos através do Q-SIM, deve-se realizar as análises em cada um dos conjuntos. O objetivo é descrever informações que



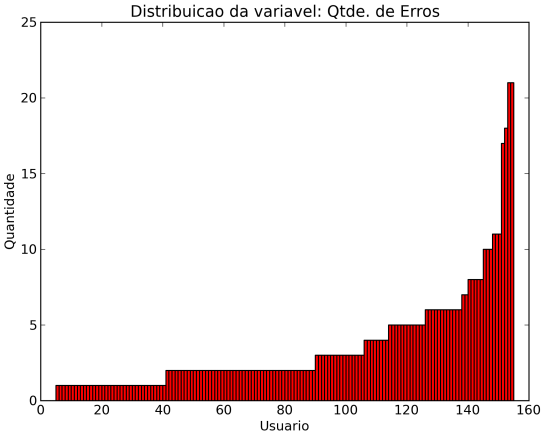
(a) Tempo de Preenchimento



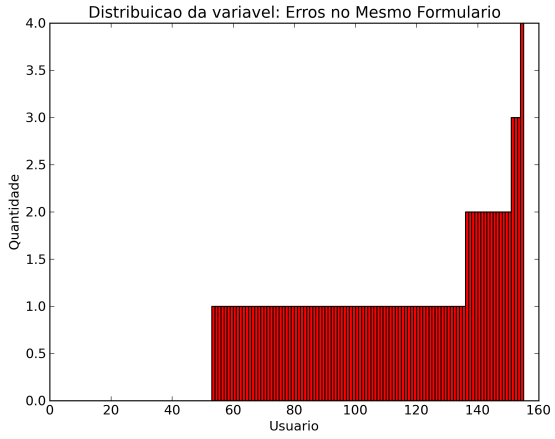
(b) Velocidade de Digitação



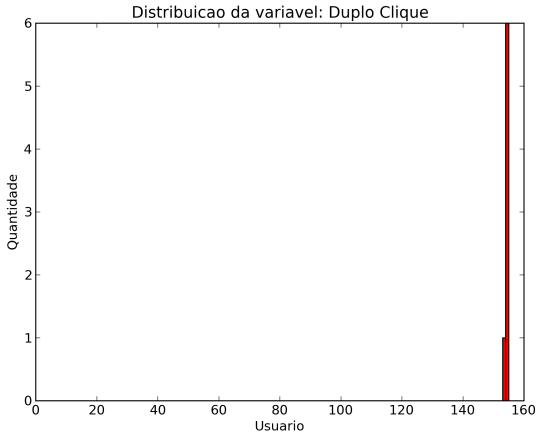
(c) Uso da tecla *Backspace*



(d) Quantidade de erros no formulário



(e) Erros reincidentes nos formulários



(f) Duplo Clique

Figura 6.1 – Distribuição dos dados coletados durante o uso do sistema

caracterizam as Personas. Cada persona deve conter informações que transmitam suas as necessidades, motivações, objetivos, habilidades, entre outras. Além das informações mencionadas, cada persona deve conter um nome e uma figura ou foto que a caracterize e facilite a comu-

nicação da equipe. As tabelas a seguir representam as cinco Personas criadas para o projeto PEAD-PMPT.

Tabela 6.3 – Persona 1: Dr. Dráuzio

Foto:	
Nome:	Dr. Dráuzio
Descrição:	Médico, aos seus 43 anos, é responsável pelo setor de inovação de um grande hospital. Entusiasta de tecnologia, gosta de passar seu tempo criando sistemas automatizados para suas pesquisas. Procura estudar sobre quais os tipos de tecnologias são mais vantajosas para melhorar o desempenho de seu trabalho. Muito preocupado com a segurança das informações e quem tem acesso à elas. Passa horas desenvolvendo aplicativos mais simples em ambientes como Access e utiliza muito contato via e-mail. Com muita experiência em utilizar sistemas desenvolvidos para *desktops*, acaba utilizando ações desses sistemas em outros tipos de sistemas, como por exemplo, sistemas web. Esse comportamento acaba gerando alguns erros na interação com os novos sistemas, mas mantém uma baixa taxa de erros na utilização do sistema.

Dados obtidos pelo Q-SIM					
Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
1.31 toques/seg	5.49 seg	1.47%	3 erros	1 vez	1 duplo clique

Para a criação das cinco Personas apresentadas nessa sessão, foram utilizados informações sobre as observações dos usuários durante testes, entrevistas com os potenciais usuários do sistema e as informações geradas pelo Q-SIM, como podemos ver nas Personas 6.3 e 6.6. As informações em vermelho corresponde as descrições que foram adicionadas com base na análise da informação gerada com o Q-SIM.

Além disso, as informações geradas com o Q-SIM são mantidas nas Personas, pois o processo pode ser realizado em um outro momento da vida do sistema e adicionadas na tabela. Assim, é possível mapear a evolução da persona ao longo do tempo e identificar a sua morte na utilização do sistema, e também o nascimento de novas Personas.

As Personas resultantes servirão de estudo para melhorar a comunicação da interface e ainda identificar componentes que facilitem e aumentem a produtividade dos usuários. Dessa forma, pode-se analisar a evolução da vida dos Personas de acordo com as melhorias realizadas no sistema. O trabalho para consumir o conhecimento das Personas está em andamento dentro do grupo de pesquisa do Laboratório de Engenharia de Usabilidade da FEI.

Dessa maneira, pode-se concluir que a metologia de criação de Personas apoiada pelo Q-SIM é capaz de gerar as Personas de maneira automatizada, através de grupos que mantém

Tabela 6.4 – Persona 2: Dra. Manuela

Foto:	
Nome:	Dra. Manuela
Descrição:	Com seus 35 anos não é muito adepta a tecnologia, utiliza o computador sempre fora do ambiente de trabalho, para enviar e-mail e navegar em redes sociais para manter contato com alguns amigos. Para suas pesquisas e trabalho prefere fazer tudo em papel, acreditando ser mais produtivo seu dia a dia. Entretanto, utiliza muito seu celular para acessar algumas coisas rápidas na internet, principalmente com o foco de manter contato com as demais pessoas, mas encontra muita dificuldade em ler as informações já que possui uma pequena deficiência visual e necessita de utilizar óculos para melhorar sua visão. Quando há necessidade de utilizar o computador em seu ambiente de trabalho procura ser atenciosa, revisar todos os seus textos antes de inserir essa informação no sistema. Dessa forma, comete poucos erros na utilização do sistema.

Dados obtidos pelo Q-SIM					
Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
1.11 toques/seg	11.46 seg	23.22%	2 erros	1 vez	0

a qualidade na similaridade entre os perfis coletados e ainda sem ruídos causados por fatores psicológicos do usuário durante uma observação ou entrevista.

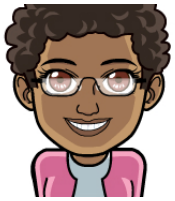
Tabela 6.5 – Persona 3: Jussara

Foto:	
Nome:	Jussara
Descrição:	Assistente Administrativa, 29 anos, fica sempre ligada no computador onde trabalha com diversos programas de fundações externas ao hospital, porém as informações produzidas por estes programas são muito essenciais para o dia a dia de trabalho dos administradores e médicos diretores do hospital. Além de manipular os programas de trabalho, ela passa muito tempo utilizando redes sociais e e-mail, dentro e fora de seu trabalho. Consegue manipular com um nível computacional avançado as planilhas eletrônicas, confeccionando relatórios para os seus superiores. Devido as diversas tarefas simultâneas, comete alguns erros nos formulários e também durante a digitação das informações. Está sempre disposta a aprender novas tarefas e ferramentas.

Dados obtidos pelo Q-SIM

Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
1.46 toques/seg	6.41 seg	48.52%	5 erros	1 vez	0

Tabela 6.6 – Persona 4: Rosana

Foto:	
Nome:	Rosana
Descrição:	Enfermeira, 45 anos, tem dificuldades em utilizar o computador, devido ao grau avançado de seu problema na visão. O seu maior contato com tecnologia são os celulares, onde utilizam apenas serviços de voz e mensagem de texto, esse ultimo com muita dificuldade devido ao tamanho das fontes serem pequenas. Contudo, é muito esforçada e gosta sempre de aprender as novidades do trabalho e lazer, mesmo com o pouco tempo para essas atividades. Ao utilizar o sistema no trabalho comete muitos erros ao preencher formulários, mas tenta sempre melhorar sua habilidade em utilizar o sistema.

Dados obtidos pelo Q-SIM

Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
1.21 toques/seg	4.17 seg	8.33%	16 erros	2 vezes	0

Tabela 6.7 – Persona 5: Ronaldo

Foto:	
Nome:	Ronaldo
Descrição:	Trabalha como assistente administrativo dentro de um grande hospital público, aos seus 60 anos, não possui nenhuma habilidade em operar produtos tecnológicos. Possui um problema de audição, entretanto esse problema não interfere em seu trabalho em seu dia a dia. Devido a sua experiência e longos anos de trabalho conhece muito bem todo o processo realizado no hospital, e consegue localizar documentos de uma maneira muito rápida agilizando o trabalho de outros profissionais dentro do hospital. Não tem problemas em aprender novas tarefas. Quando utiliza o computador possui muita dificuldade em digitar e erra muito durante o processo de digitação nos campos. Devido a sua pressa para agilizar o seu trabalho, acaba cometendo alguns erros no formulário, mas consegue realiza-lo com sucesso.

Dados obtidos pelo Q-SIM

Velocidade Digitação	Tempo de Preenchimento	Uso do Backspace	Qtde. de Erros	Erros no mesmo Formulário	Duplo Clique
0.92 toques/seg	5.94 seg	75%	7 erros	1 vez	0

7 CONCLUSÕES E TRABALHOS FUTUROS

Os resultados obtidos durante os testes do algoritmo Q-SIM foram satisfatórios. Ele demonstrou que consegue obter resultados melhores e/ou semelhantes aos algoritmos utilizados na área de *data clustering*, como o k-*means*, o DBSCAN e o *Affinity Propagation*. Esses algoritmos podem demonstrar superioridade em alguns tipos de tarefas. Entretanto, um dos possíveis trabalhos futuros como sequência desse trabalho é demonstrar que o Q-SIM pode ser utilizado não só em tarefas de modelagem de usuário, mas também em outros tipos de *data clustering* com um desempenho semelhante aos demais algoritmos.

O desempenho computacional do Q-SIM é similar aos demais algoritmos, mas ele depende muito da quantidade de elementos que cada grupo possui, além da quantidade de grupos formadas ao longo do processo. Como nenhum processo de otimização foi realizado para o Q-SIM, a comparação do desempenho computacional não foi realizada nesse momento, mas será tratada em um trabalho futuro. Entretanto, para o problema com agrupamento de perfis de usuários similares o Q-SIM demonstrou que obtêm o grupos mais similares e com mais qualidade, nos grupos formados, do que os demais algoritmos.

Dessa forma, ele se torna a melhor opção para o especialista em modelagem de usuários, principalmente para a técnica de Personas, quando o foco dele é manter o grau de similaridade entre os elementos do grupo. Essa característica é muito importante na modelagem de usuários, pois quanto mais similares forem os modelos mais próximo de representar o usuário real este modelo estará. Além disso, é possível que o especialista varie o valor de similaridade do Q-SIM e verificar a quantidade de grupos formados, caso seja necessário aumentar ou diminuir o número de grupos final. Contudo, o Q-SIM sempre manterá a similaridade entre os elementos do grupo, de acordo com o solicitado pelo especialista.

Quando apoiado pelo Q-SIM, o processo de criação de personas de maneira automatizada proposta por essa dissertação, tornou-se possível criar modelos significativos para análise da interface e melhoria da mesma. Aplicando o processo no projeto PEAD-PMPT, as personas obtidas são capazes de demonstrar que existe uma barreira na comunicação da interface do sistema PEAP-PMPT, uma vez que o número de erros entre as personas é alto. Os modelos são gerados com o conhecimento construído na utilização do sistema em funcionamento, portanto são mais utilizados para conhecimento em novos projetos e para melhoria do sistema atual.

Além do mais, esse processo de criação de personas pode gerar insumo para um componente de interface adaptativa, ao uni-lo com sistema de padrões para interfaces. Os estudos estão encaminhados junto ao grupo de pesquisa do Laboratório de Engenharia de Usabilidade da FEI, para que o componente de interface adaptativa seja concretizado.

A pesquisa realizada na dissertação possibilitou a abertura para diversos outros trabalhos e pesquisa. Os trabalhos relacionados ao algoritmo Q-SIM são muitos, como por exemplo, um estudo da melhoria no desempenho computacional que impacta diretamente na complexidade

do algoritmo e no consumo de memória do mesmo. Outro trabalho que deve ser continuado é a aplicação do Q-SIM em diferentes tipos de aplicação de *clustering*, e realizada a comparação com os algoritmos para cada tipo de aplicação, como por exemplo sistemas de recomendação onde pode-se identificar os padrões nas recomendações realizadas e também PICAPS, apresenta por Aquino Junior (2008).

Os trabalhos futuros aplicados a IHC e a modelagem de usuários também são diversos. As variáveis mapeadas podem ser ampliadas em sua quantidade na busca de um melhor entendimento das habilidades, comportamento e experiência do usuário na utilização do sistema. Um estudo sobre o comportamento do usuário para determinadas tarefas podem ser analisados com a ação do clique nos componentes da interface, desenvolvendo uma melhor navegação e atalhos para usuários de maneira personalizada. O consumo dos conhecimentos gerados com base na análise das personas obtidas, sendo que esse consumo pode ser realizado de maneira automática ou manualmente através da análise do especialista, também é um dos possíveis trabalhos futuros observados nos resultados obtidos. Por fim, o estudo sobre a evolução das personas ao longo do tempo, identificando o tempo de vida e se essa evolução é positiva ou não.

REFERÊNCIAS

AQUINO JUNIOR, P. T. **PICaP:**: padrões e personas para expressão da diversidade de usuários no projeto de interação. Tese (Doutorado), São Paulo, Brasil, 2008. Biblioteca Digital de Teses e Dissertações da USP. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-15092008-144412/>.

AQUINO JUNIOR, P. T.; FILGUEIRAS, L. V. L. User modeling with personas. In: **Proceedings of the 2005 Latin American conference on Human-computer interaction**. New York: ACM, 2005. (CLIHC '05), p. 277–282. Disponível em: <http://doi.acm.org/10.1145/1111360.1111388>.

AQUINO JUNIOR, P. T.; FILGUEIRAS, L. V. L. A expressão da diversidade de usuários no projeto de interação com padrões e personas. In: **Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems**. Porto Alegre: Sociedade Brasileira de Computação, 2008. (IHC '08), p. 1–10. Disponível em: <http://dl.acm.org/citation.cfm?id=1497470.1497472>.

ATZENI, A. et al. Here's johnny: A methodology for developing attacker personas. In: **Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security**. Washington: IEEE Computer Society, 2011. (ARES '11), p. 722–727. Disponível em: <http://dx.doi.org/10.1109/ARES.2011.115>.

BARBOSA, S. D. J.; SILVA, B. S. d. **Interação Humano-Computador**. [S.l.]: Campus-Elsevier, 2010.

BEZDEK, J.; PAL, N. Cluster validation with generalized dunn's indices. In: **Artificial Neural Networks and Expert Systems, 1995. Proceedings., Second New Zealand International Two-Stream Conference on**. [S.l.: s.n.], 1995. p. 190 –193.

BLACK, P. E. **Dictionary of algorithms and data structures**. [S.l.]: National Institute of Standards and Technology, 2004.

BRICKEY, J.; WALCZAK, S.; BURGESS, T. Comparing semi-automated clustering methods for persona development. **Software Engineering, IEEE Transactions on**, IEEE, n. 99, p. 1–1, 2011.

COOPER, A. **The Inmates Are Running the Asylum**. Indianapolis: Macmillan Publishing Co., Inc., 1999.

COOPER, A.; REIMANN, R.; CRONIN, D. **About face 3: the essentials of interaction design**. New York: John Wiley & Sons, Inc., 2007.

D'ANGELO, F. d. M. **Identificação automática de perfis de grupos de usuários de interfaces WEB**. São Bernardo do Campo, São Paulo, Brasil: Dissertação (Mestrado em Engenharia Elétrica) - Centro Universitário da FEI, 2012. Biblioteca Digital de Teses e Dissertações da FEI. Disponível em: <http://tede.fei.edu.br/tede/tde-busca/arquivo.php?codArquivo=231/>.

DASGUPTA, A. et al. Overcoming browser cookie churn with clustering. In: **Proceedings of the fifth ACM international conference on Web search and data mining**. New York: ACM, 2012. (WSDM '12), p. 83–92. Disponível em: <http://doi.acm.org/10.1145/2124295.2124308>.

DAVIES, D. L.; BOULDIN, D. W. A cluster separation measure. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, v. 1, n. 2, p. 224–227, fev. 1979. Disponível em: <http://dx.doi.org/10-.1109/TPAMI.1979.4766909>.

DEZA, M. M.; DEZA, E. **Encyclopedia of distances**. [S.l.]: Springer, 2009.

DUTTA, M.; MAHANTA, A. K.; PUJARI, A. K. Qrock: A quick version of the rock algorithm for clustering of categorical data. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, v. 26, n. 15, p. 2364–2373, nov. 2005. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2005.04.008>.

ESTELL, J.; REID, K. Work in progress - development of personas: Emphasizing human need in a first-year engineering capstone course. In: **Frontiers in Education Conference (FIE), 2010 IEEE**. [S.l.: s.n.], 2010. p. T4E–1 –T4E–2.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: . [S.l.]: AAAI Press, 1996. p. 226–231.

FAILY, S.; FLECHAIS, I. Persona cases: a technique for grounding personas. In: **Proceedings of the 2011 annual conference on Human factors in computing systems**. New York: ACM, 2011. (CHI '11), p. 2267–2270. Disponível em: <http://doi.acm.org/10.1145/1978942.1979274>.

FILGUEIRAS, L. et al. Personas como modelo de usuários de serviços de governo eletrônico. In: **Proceedings of the 2005 Latin American conference on Human-computer interaction**. New York: ACM, 2005. (CLIHC '05), p. 319–324. Disponível em: <http://doi.acm.org/10.1145/1111360.1111395>.

FREY, B. J.; DUECK, D. Clustering by passing messages between data points. **Science**, v. 315, n. 5814, p. 972–976, 2007. Disponível em: <http://www.sciencemag.org/content/315/5814/972.abstract>.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability; A Guide to the Theory of NP-Completeness**. New York: W. H. Freeman & Co., 1990.

GUO, J.; YAN, P. User-centered information architecture of university library website. In: IEEE. **Computer Research and Development (ICCRD), 2011 3rd International Conference on**. [S.l.], 2011. v. 2, p. 370–372.

JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, v. 31, n. 8, p. 651–666, jun. 2010. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2009.09.011>.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Comput. Surv.**, ACM, New York, v. 31, n. 3, p. 264–323, set. 1999. Disponível em: <http://doi.acm.org/10.1145/331499.331504>.

KAN, W. et al. Personas construction based on utility analysis in industrial design. In: **Management and Service Science (MASS), 2010 International Conference on**. [S.l.: s.n.], 2010. p. 1 –4.

KANTARDZIC, M. **Data mining: concepts, models, methods, and algorithms**. [S.l.]: Wiley-IEEE Press, 2011.

KOVÁCS, F.; LEGÁNY, C.; BABOS, A. Cluster validity measurement techniques. In: CITESEER. **6th International Symposium of Hungarian Researchers on Computational Intelligence**. [S.l.], 2005.

LATTIN, J.; CARROL, D.; GREEN, P. **Análise de dados multivariados**. [S.l.]: São Paulo: Cegrage Learning, 2011.

LEGáNY, C.; JUHáSZ, S.; BABOS, A. Cluster validity measurement techniques. In: **Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases**. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2006. (AIKED'06), p. 388–393. Disponível em: <http://dl.acm.org/citation.cfm?id=1364262.1364328>.

LOYOLA, P.; ROMÁN, P.; VELÁSQUEZ, J. Clustering-based learning approach for ant colony optimization model to simulate web user behavior. In: IEEE COMPUTER SOCIETY. **Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01**. [S.l.], 2011. p. 457–464.

MAHALANOBIS, P. C. On the generalized distance in statistics. In: NEW DELHI. **Proceedings of the National Institute of Sciences of India**. [S.l.], 1936. v. 2, n. 1, p. 49–55.

MASIERO, A. A. et al. Multidirectional knowledge extraction process for creating behavioral personas. In: **Proceedings of the 10th Brazilian Symposium on on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction**. Porto Alegre: Brazilian Computer Society, 2011. (IHC+CLIHC '11), p. 91–99. Disponível em: <http://dl.acm.org/citation.cfm?id=2254436.2254454>.

MATTHEWS, T. et al. Collaboration personas: a new approach to designing workplace collaboration tools. In: **Proceedings of the 2011 annual conference on Human factors in computing systems**. New York: ACM, 2011. (CHI '11), p. 2247–2256. Disponível em: <http://doi.acm.org/10.1145/1978942.1979272>.

MEISSNER, F.; BLAKE, E. Understanding culturally distant end-users through intermediary-derived personas. In: **Proceedings of the South African Institute of Computer Scientists and Information Technologists Conference on Knowledge, Innovation and Leadership in a Diverse, Multidisciplinary Environment**. New York: ACM, 2011. (SAICSIT '11), p. 314–317. Disponível em: <http://doi.acm.org/10.1145/2072221.2072266>.

MITRA, S.; ACHARYA, T. **Data Mining: multimedia, soft computing, and bioinformatics**. [S.l.]: Wiley-Interscience, 2003.

MUHLENBACH, F.; LALLICH, S. A new clustering algorithm based on regions of influence with self-detection of the best number of clusters. In: **Proceedings of the 2009 Ninth IEEE International Conference on Data Mining**. Washington: IEEE Computer Society, 2009. (ICDM '09), p. 884–889.

NG, R. T.; HAN, J. Efficient and effective clustering methods for spatial data mining. In: **Proceedings of the 20th International Conference on Very Large Data Bases**. San Francisco: Morgan Kaufmann Publishers, 1994. (VLDB '94), p. 144–155. Disponível em: <http://dl.acm.org/citation.cfm?id=645920.672827>.

NUNES, F.; SILVA, P. A.; ABRANTES, F. Human-computer interaction and the older adult: an example using user research and personas. In: **Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments**. New York: ACM, 2010. (PETRA '10), p. 49:1–49:8. Disponível em: <http://doi.acm.org/10.1145/1839294.1839353>.

ONU. **World population ageing 1950-2050**. [S.l.]: UN, 2002.

PRUITT, J.; ADLIN, T. **The Persona Lifecycle: Keeping People in Mind Throughout Product Design**. San Francisco: Morgan Kaufmann Publishers, 2005.

PUTNAM, C.; KOLKO, B.; WOOD, S. Communicating about users in ictd: leveraging hci personas. In: **Proceedings of the Fifth International Conference on Information and Communication Technologies and Development**. New York: ACM, 2012. (ICTD '12), p. 338–349. Disponível em: <http://doi.acm.org/10.1145-/2160673.2160714>.

SAEZ, A. V.; DOMINGO, M. G. Scenario-based persona: introducing personas through their main contexts. In: TAN, D. S. et al. (Ed.). **Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Extended Abstracts Volume, Vancouver, BC, Canada, May 7-12, 2011**. [S.l.]: ACM, 2011. p. 505.

SMYTH, B.; MCKENNA, E. Competence guided incremental footprint-based retrieval. **Knowledge-Based Systems**, v. 14, p. 155 – 161, 2001. Disponível em: <http://www.sciencedirect.com/science/article/pii-/S0950705101000922>.

TU, N. et al. Using cluster analysis in persona development. In: IEEE. **Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on**. [S.l.], 2010. p. 1–5.

TURNER, P.; TURNER, S. Is stereotyping inevitable when designing with personas? **Design Studies**, v. 32, n. 1, p. 30 – 44, 2011. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0142694X10000451>.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. 3rd. ed. San Francisco: Morgan Kaufmann Publishers, 2011.

XIAOMING, D.; XIAOYAN, M. A web users clustering model based on users' browsing path. In: IEEE. **Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on**. [S.l.], 2009. p. 1–4.