4
Aplicativo para Análise de Agrupamentos

Este capítulo apresenta a modelagem de um aplicativo, denominado *Cluster Analysis*, dedicado à formação e análise de grupos em bases de dados.

O aplicativo desenvolvido tem como objetivo principal facilitar a tarefa de análise de agrupamento de dados, através da disponibilização de diversos métodos já consagrados. Além disso, ele é flexível o suficiente para permitir a incorporação de outros métodos, inclusive aqueles que porventura venham a ser desenvolvidos por usuários.

O aplicativo foi desenvolvido em Java e atualmente disponibiliza para o usuário métodos de agrupamento de dados internos, presentes no aplicativo, e métodos provenientes de interface com dois softwares comerciais: Matlab® e R®, como pode ser observado no diagrama da Figura 13.

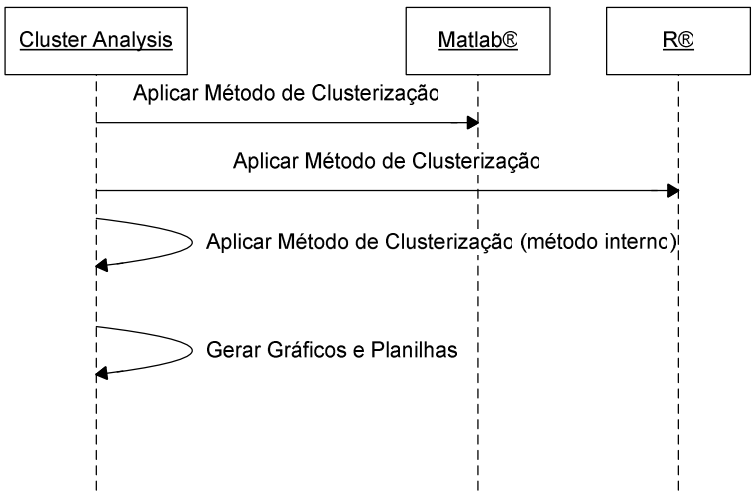

Figura 13: Diagrama de Seqüência - Interface com Matlab® e R®.

Foram desenvolvidos funções e scripts em Matlab® e no R® para a utilização adequada de cada um dos seus métodos de agrupamento de dados, carregando de forma apropriada a base de dados selecionada e gerando a entrada esperada pelo aplicativo.

No aplicativo desenvolvido foram utilizados os mesmos nomes dos métodos usados nos aplicativos externos, a fim de permitir o usuário identificar rapidamente o método empregado.

Os seguintes métodos de agrupamento de dados disponíveis no Matlab® estão presentes no aplicativo:

- *Métodos Hierárquicos Aglomerativos*
  - ⇒ **Linkage** (nome do método aglomerativo como está no Matlab®)
    - Métodos de Distância entre Grupos
      - Weighted
      - Centróide
      - Ligação Completa
      - Ligação Simples
      - Média das Ligações
      - Ward

- *Métodos Particionais*
  - ⇒ **K-Means**
  - ⇒ **Fuzzy C-Means**

Os seguintes métodos de agrupamento de dados disponíveis no R® estão presentes no aplicativo:

- *Métodos Hierárquicos Aglomerativos*
  - ⇒ **AGNES**
    - Métodos de Distância entre Grupos
      - Weighted
      - Ligação Completa
      - Ligação Simples
      - Média das Ligações
      - Ward

- *Métodos Hierárquicos Divisivos*
  - ⇒ **DIANA**

- *Métodos Particionais*
  - ⇒ **PAM**
  - ⇒ **CLARA**
  - ⇒ **FANNY**

Além de poder utilizar os modelos de agrupamento presentes no Matlab e no R, o aplicativo também implementa dois métodos particionais de agrupamento de dados: o **K-Means** e o **Fuzzy C-Means**.

O aplicativo gera como resultado duas planilhas e até três gráficos, dependendo do método escolhido, como segue:

- *Tabela de Pertinências*

  Tabela contendo os valores de pertinência dos dados aos agrupamentos.
- *Tabela de Médias*

  Tabela com as médias dos valores de cada atributo por agrupamento.
- *Gráfico da Silhueta*
- *Dendograma (apenas para métodos hierárquicos)*
- *Gráfico Comparativo (opcional)*

  Gráfico de pizza por agrupamento entre dois atributos do conjunto de dados.

As planilhas e gráficos serão mais bem descritas na próxima seção, que abordará também os processos do aplicativo.

4.1.
Modelagem do Aplicativo

A modelagem do aplicativo pode ser dividida em quatro blocos principais, conforme apresentado na Figura 14.

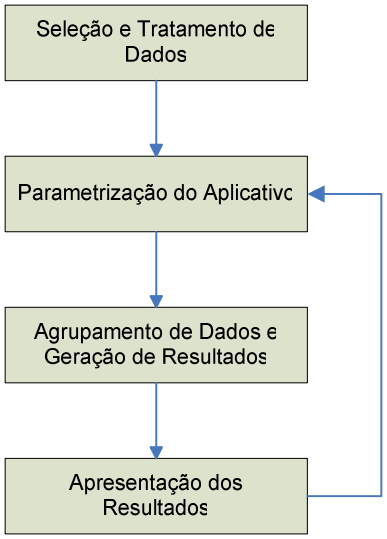

Figura 14: Diagrama de Blocos: Visão geral do processo e aplicativo para análise de grupos.

⇒ *Seleção e Tratamento de Dados*

Processo responsável pelo pré-processamento dos dados.

⇒ *Parametrização do Aplicativo*

Processo responsável pela parametrização do processo de agrupamento de dados, bem como os resultados a serem analisados.

⇒ *Agrupamento de Dados e Geração de Resultados*

Processo responsável pelo agrupamento de dados, bem como a geração dos dados a serem utilizados na etapa de apresentação dos resultados.

⇒ *Apresentação dos Resultados*

Etapa responsável pelo processamento dos dados da etapa anterior para geração dos gráficos e planilhas.

Cada módulo está descrito detalhadamente nas subseções seguintes.

4.1.1.
Seleção e Tratamento de Dados

Esse processo pode ser dividido em cinco blocos principais, conforme apresentado na Figura 15.

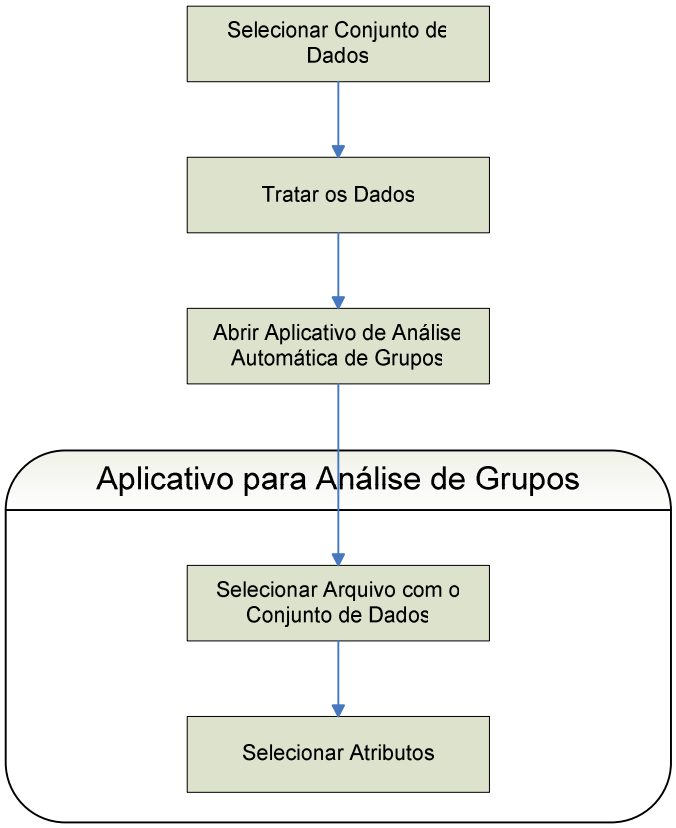

Figura 15: Diagrama de Blocos: Visão detalhada do processo de Seleção e Tratamento de Dados.

O processo de análise de agrupamento de dados inicia-se com a seleção e preparação do conjunto de dados a ser utilizado na fase de agrupamento. Estas duas etapas envolvem a seleção de um conjunto de dados representativo da base de dados e a realização de processos de tratamento de dados, a fim de garantir a qualidade dos dados que serão utilizados nas etapas seguintes do processo.

As etapas descritas abaixo se referem a como essas tarefas são realizadas na estrutura do aplicativo:

- *Selecionar Arquivo com o Conjunto de Dados*

  Etapa onde o usuário deverá selecionar a base de dados a ser usada pelo aplicativo.

- *Selecionar Atributos*

  Nessa etapa são apresentados todos os atributos encontrados na base de dados selecionada. O usuário deverá selecionar os atributos que farão parte do processo de agrupamento de dados, ou seja, os atributos que possam sugerir alguma relevância ao processo.

A Figura 16 mostra a interface de como os dois processos acima são realizados no aplicativo.

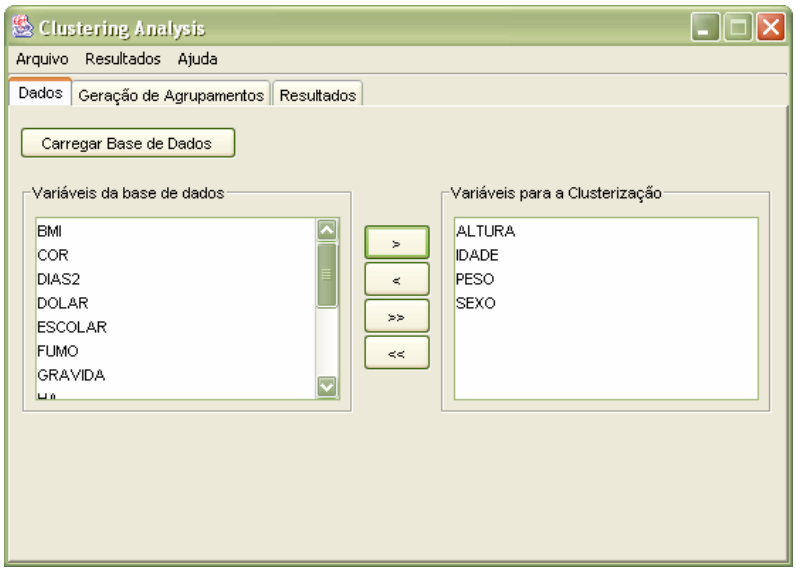

Figura 16: Aplicativo de Análise de Grupos: Seleção e Tratamento de Dados.

4.1.2. Parametrização do Aplicativo

Esse processo pode ser dividido em dois blocos principais, conforme apresentado na Figura 17.

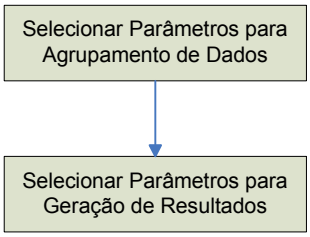

Figura 17: Diagrama de Blocos: Visão detalhada do processo de Parametrização do Aplicativo.

Essa etapa é composta por dois processos que são responsáveis pela parametrização do aplicativo, como segue:

- *Selecionar Parâmetros para Agrupamento de Dados*

  Nessa etapa o usuário define os parâmetros que serão utilizados no agrupamento de dados: o número de agrupamentos e o método a ser utilizado no processo.

- *Selecionar Parâmetros para Geração de Resultados*

  Nessa etapa o usuário define os parâmetros que serão utilizados na geração de resultados: os atributos que serão utilizados na geração da tabela de média e os atributos que serão utilizados para montar o gráfico comparativo.

A Figura 18 mostra a interface do aplicativo onde ocorre a etapa de seleção de parâmetros para agrupamento de dados.

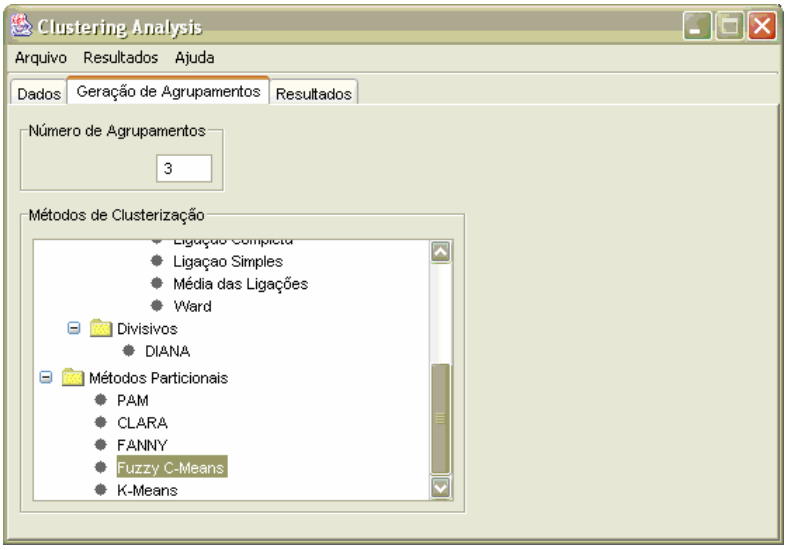

Figura 18: Aplicativo de Análise de Grupos: Geração de Agrupamentos

A Figura 19 mostra a interface do aplicativo onde ocorre a etapa de seleção de parâmetros para geração de resultados.

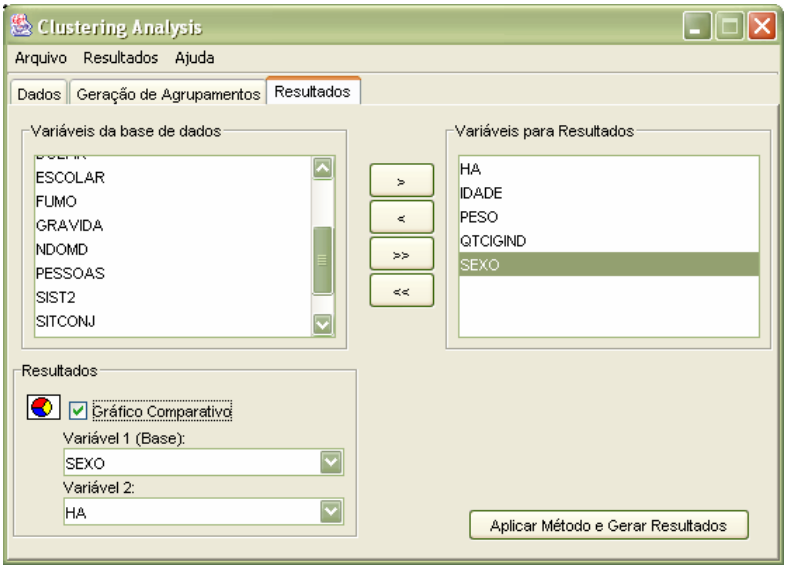

Figura 19: Aplicativo de Análise de Grupos: Geração de Resultados.

4.1.3. Agrupamento de Dados e Geração de Resultados

Esse processo pode ser dividido em onze blocos principais, conforme apresentado na Figura 20.

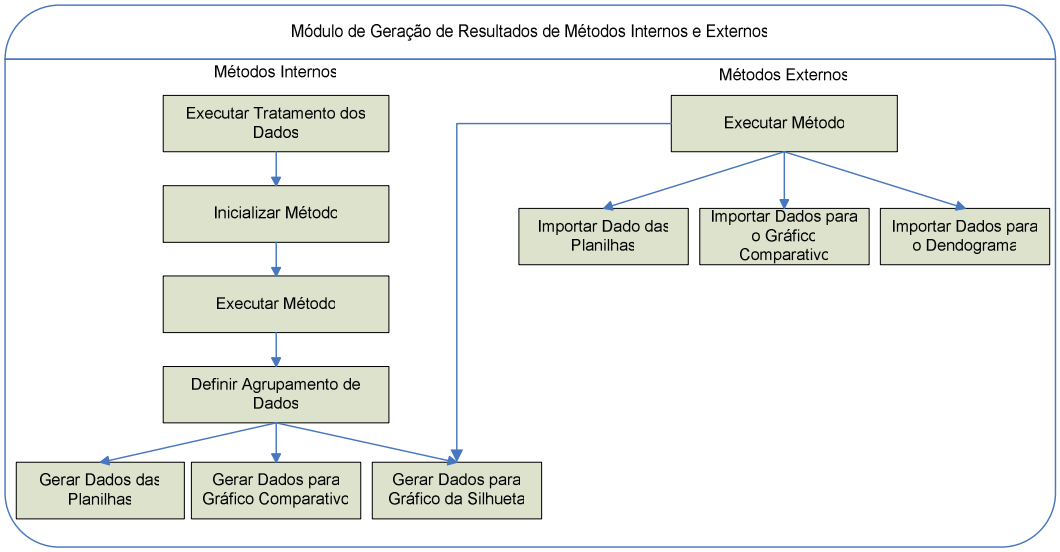

Figura 20: Diagrama de Blocos: Visão detalhada do processo de Agrupamento de Dados e Geração de Resultados

Essa etapa é responsável primeiramente por identificar a que grupo o método selecionado pertence: métodos internos ou externos.

Depois de identificado, o método é executado de acordo com os módulos abaixo.

⇒ *Métodos Internos*

- *Executar Tratamento dos Dados*

  Nessa etapa os dados são carregados no sistema e normalizados.

- *Inicializar Método*

  Nessa etapa é identificado o método e inicializado os parâmetros necessários para a sua execução.

- *Executar Método*

  Nessa etapa dá-se início à execução do método e à construção da matriz contendo os valores de pertinência dos dados aos agrupamentos.

- *Definir Agrupamento de Dados*

  Nessa etapa é criada a matriz que define a que agrupamento os dados pertencem, a partir da análise da matriz contendo os valores de pertinência dos dados aos agrupamentos.

- *Gerar Dados das Planilhas*

  Nessa etapa os dados para a tabela de pertinências já foram construídos pelo processo de execução do método. Para a geração dos dados da tabela de média são processados os valores de média para cada agrupamento.

- *Gerar Dados para o Gráfico Comparativo*

  Nessa etapa, primeiramente são localizados os valores distintos para cada um dos dois atributos definidos no processo de parametrização da geração de resultados. Em seguida, é feito um processamento para agrupar os valores correlacionados entre esses dois atributos em questão, procurando totalizar a quantidade de dados por agrupamento.

⇒ *Métodos Externos*

Essa etapa é responsável pela interface entre o aplicativo e os métodos externos.

As etapas de tratamento de dados, inicialização e execução do método devem estar presentes internamente nos métodos externos, bem como a geração dos dados que serão importados pelo aplicativo para a geração

das planilhas, do gráfico comparativo, do dendograma e do gráfico da silhueta.

- *Importar Dados das Planilhas*

  Essa etapa é responsável pela importação dos dados para as tabelas de pertinências e de média.

- *Importar Dados para o Gráfico Comparativo*

  Essa etapa é responsável pela importação dos dados que serão utilizados para a geração do gráfico comparativo.

- *Importar Dados para o Dendograma*

  Essa etapa é responsável pela importação dos dados que serão utilizados para a geração do dendograma.

- *Gerar Dados para Gráfico da Silhueta*

  Nessa etapa são calculados os valores de silhueta para cada dado em cada agrupamento.

4.1.4. Apresentação dos Resultados

Esse processo pode ser dividido em quatro blocos principais, conforme apresentado na Figura 21.

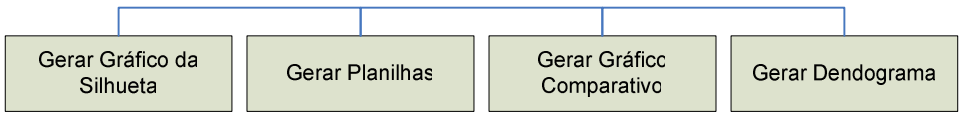

Figura 21: Diagrama de Blocos: Visão detalhada do processo de Apresentação dos Resultados.

Neste passo são disponibilizados as tabelas e gráficos que servem de auxílio à análise dos resultados gerados pelo processo de agrupamento de dados. Os processo são descritos a seguir.

- *Gerar Gráfico da Silhueta*

  O gráfico da silhueta é a representação gráfica da qualidade dos agrupamentos gerados.

  Essa etapa é responsável pela análise, processamento e geração do gráfico da silhueta.

A Figura 22 mostra a interface do aplicativo na geração do gráfico da silhueta.

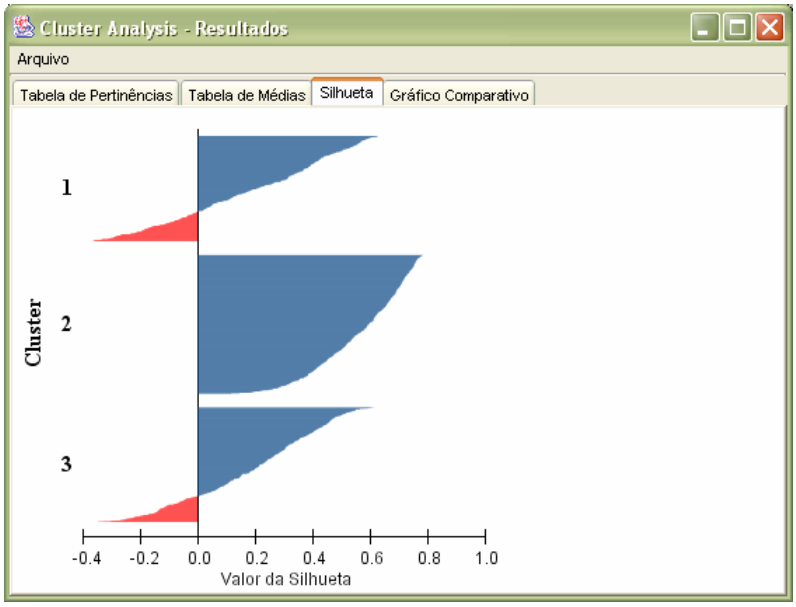

Figura 22: Aplicativo de Análise de Grupos: Gráfico da Silhueta.

- *Gerar Planilhas*

  Essa etapa é responsável pela geração das tabelas de pertinência e de média, a partir dos dados gerados pelos métodos internos ou importado após execução dos métodos externos.

  - *Tabela de Pertinências*

    Essa tabela contém os graus de pertinência de cada dado a cada agrupamento. Para métodos de agrupamento de dados fuzzy, tem-se graus de pertinência entre 0 e 1, e para os demais métodos têm-se graus de pertinência 0 ou 1.

    A tabela é formada por *(k+1)* colunas e *n* linhas, onde *k* é a quantidade de agrupamentos e *n* é a quantidade de dados.

    Acima da tabela é exibido o valor de pertinência máximo e mínimo dos dados. Esse tipo de informação é apenas útil para métodos de agrupamento de dados fuzzy, onde se pode identificar rapidamente o resultado da evolução do algoritmo.

    A Figura 23 mostra a interface do aplicativo na geração da tabela de pertinências.

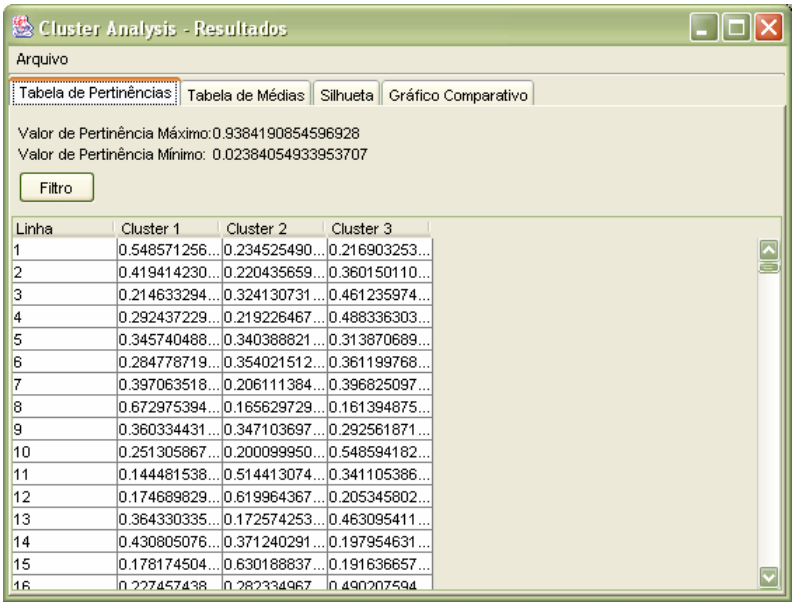

Figura 23: Aplicativo de Análise de Grupos: Tabela de Pertinências.

Para uma melhor análise dos resultados dessa planilha, há também a opção de filtro por cluster e por intervalo de valor, que é útil para a análise de desempenho de um determinado método.

- *Tabela de Médias*

  Tabela contendo a média dos valores das variáveis selecionadas para apresentação dos resultados por grupos, e total de elementos em cada agrupamento.

  A Figura 24 mostra a interface do aplicativo na geração da tabela de médias.

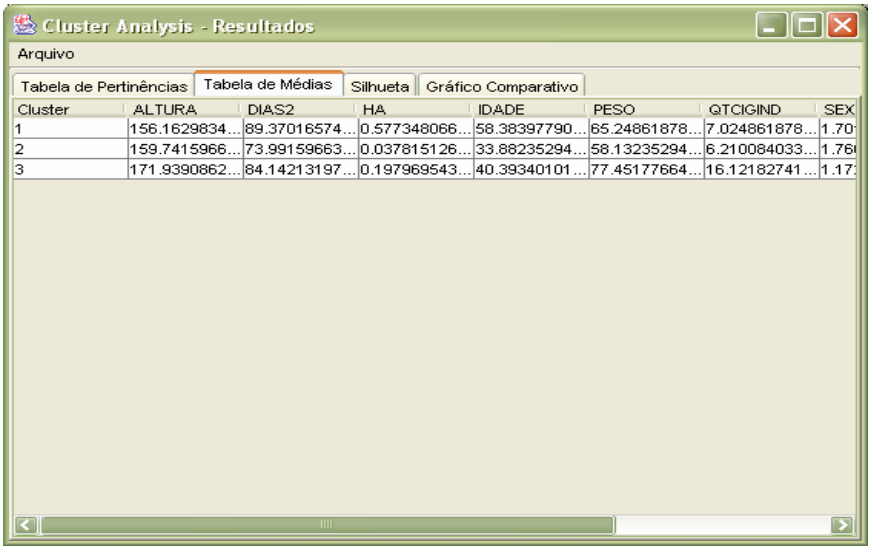

Figura 24: Aplicativo de Análise de Grupos: Tabela de Médias.

- *Gerar Gráfico Comparativo*

  O gráfico comparativo é um gráfico de pizza por agrupamento entre os dois atributos selecionados na parametrização dos resultados.

  Essa etapa é responsável pela análise, processamento e geração do gráfico comparativo, a partir dos dados gerados pelos métodos internos ou importado após execução dos métodos externos.

  A Figura 25 mostra a interface do aplicativo na geração da gráfico comparativo.

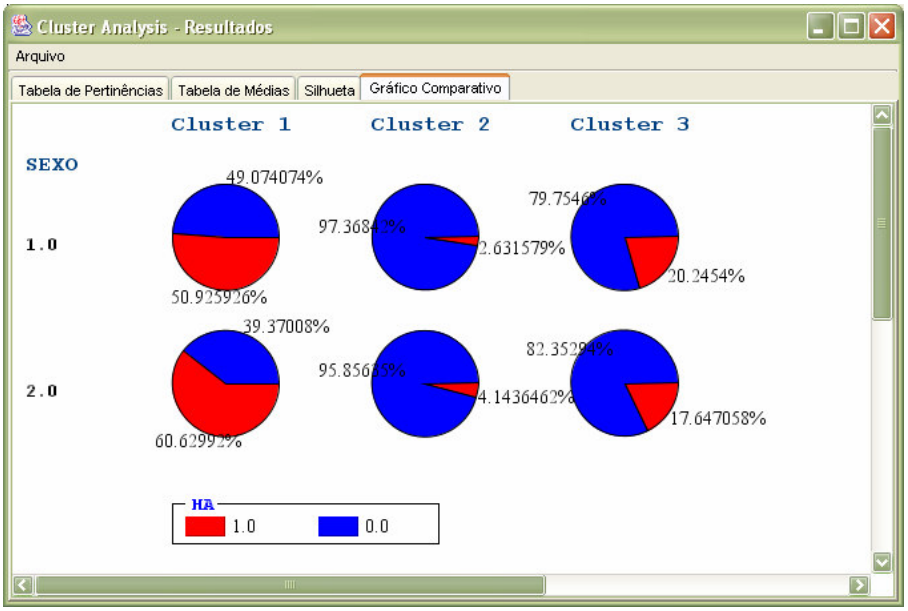

Figura 25: Aplicativo de Análise de Grupos:Gráfico Comparativo.

O exemplo acima foi retirado do estudo de caso visto no próximo capítulo. Nele foram selecionados os atributos *Sexo* como variável base e *HA* (hipertensão arterial) como a segunda variável.

O gráfico mostra o percentual de indivíduos pertencentes a cada agrupamento do sexo masculino (valor 1.0) e do sexo feminino (valor 2.0) que tem ou não hipertensão arterial (presença de hipertensão – valor 1.0 / ausência de hipertensão – valor 0.0).

- *Gerar Dendograma*

  Dendograma é o diagrama que mostra a hierarquia e a relação dos agrupamentos em uma estrutura.

Essa etapa é responsável pela análise, processamento e geração do dendograma para métodos hierárquicos aglomerativos ou divisivos.

A Figura 26 mostra a interface do aplicativo na geração do dendograma.

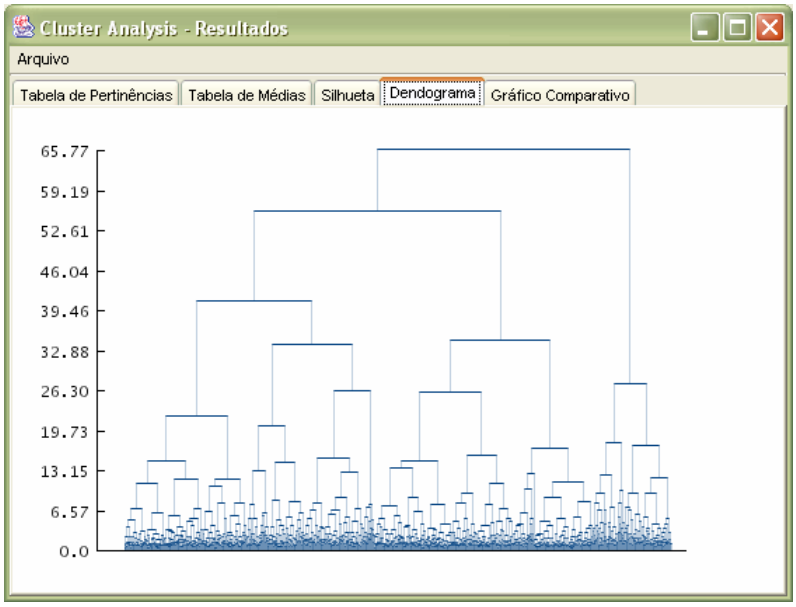

Figura 26: Aplicativo de Análise de Grupos: Dendograma.

4.1.5.
Modelagem Detalhada do Aplicativo

O processo detalhado pode ser dividido em 23 blocos principais conforme visto nas seções anteriores, apresentados a seguir.

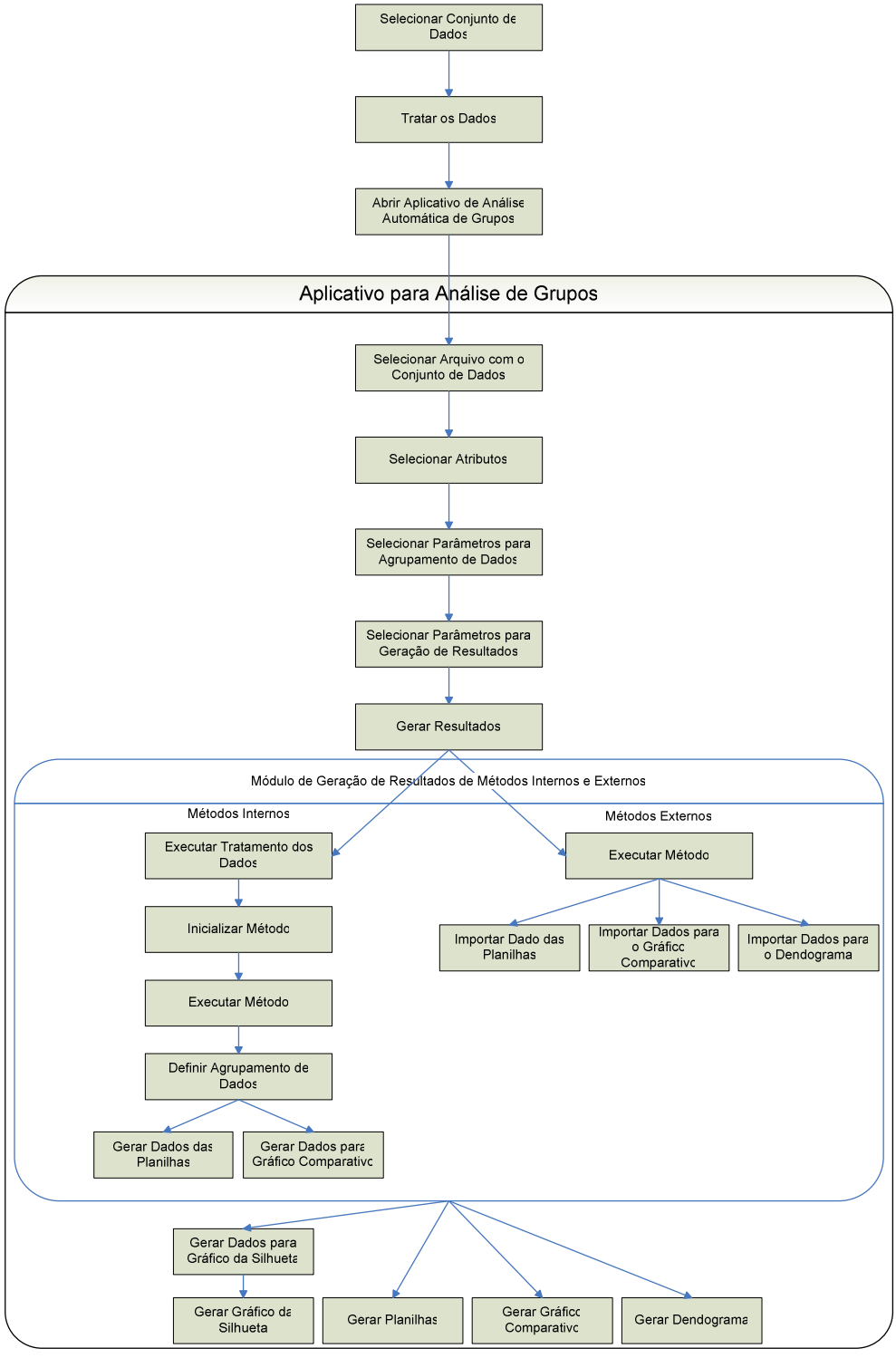

Figura 27: Diagrama de blocos detalhado do aplicativo desenvolvido

4.2. Disponibilizando outros Métodos

Atualmente o usuário pode desenvolver métodos internos, presentes no aplicativo, ou métodos externos, provenientes de interface com aplicativos externos. Caso o usuário desenvolva outros métodos de agrupamento de dados no Matlab® ou no R®, ele pode facilmente fazer uso da interface já existente; caso contrário, basta fazer alterações simples no aplicativo para a chamada adequada dos métodos, e no caso de métodos externos, criar as interfaces necessárias para que os métodos sejam chamados adequadamente.

4.2.1. Métodos Externos

Em todos os casos onde é necessária uma interface entre o aplicativo desenvolvido e outro aplicativo externo, é necessário que as implementações gerem os arquivos adequados solicitados pelo aplicativo.

Os arquivos de interface solicitados pelo aplicativo são necessários fundamentalmente para a geração dos resultados (planilhas e gráficos).

Abaixo são listados os arquivos necessários, a sua utilização e o formato como os dados devem estar disponíveis.

- *dados_mean.txt*

  Utilizado para gerar a tabela de médias.

  Formato:

  Tabela 3: Arquivo: dados_mean.txt – Formato dos dados.

<nome_atributo_1>	<nome_atributo_2>	...	Total
<média_grupo_1>	<média_grupo_1>	...	<total_grupo_1>
<média_grupo_2>	<média_grupo_2>	...	<total_grupo_2>
⋮	⋮	⋮	⋮

- *dados_idx.txt*

  Utilizado para gerar a tabela de pertinências.

  Formato:

  Tabela 4: Arquivo: dados_idx.txt – Formato dos dados.

Cluster 1	Cluser 2	...
<pertinência_dado_1>	<pertinência_dado_1>	...
<pertinência_dado_2>	<pertinência_dado_2>	...

⋮	⋮	⋮

- *dados_comparativo.txt*

  Utilizado no processamento e geração do gráfico comparativo.

  Formato:

  Tabela 5: Arquivo: dados_comparativo.txt – Formato dos dados.

<nome_atributo_1>	<nome_atributo_2>	Total.Cluster.1	Total.Cluster.2	...
Linhas com valores que a variável pode assumir	Linhas com valores que a variável pode assumir	Total de dados do grupo 1 com valores dos atributos 1 e 2.	Total de dados do grupo 2 com valores dos atributos 1 e 2.	...

- *dados_dendograma.txt*

  Utilizado no processamento e geração do dendograma.

  Formato:

  - Matriz de dimensão *(n-1)x3*, onde *n* é a quantidade de dados.
  - Coluna 1 e 2 contém os índices dos dados que foram agrupados. Este novo agrupamento formado por essa união terá como índice o valor *n+1*.
  - Coluna 3 contém os graus de similaridades entre os agrupamentos.

4.2.1.1. Métodos no Matlab®

Para criar métodos no Matlab® é necessário conhecer também um pouco da linguagem de programação Java, linguagem utilizada para desenvolver o aplicativo. Siga os seguintes passos para a criação de novos métodos:

1) Crie o novo método no Matlab® na pasta *Matlab Work*.
2) Adicione o novo método criado dentro do método *executeSegmenta.m*. Procure seguir o mesmo padrão adotado nos outros métodos como chamada e retorno esperado do método.
3) Dentro do Matlab® execute o seguinte comando:

mcc -m -I '<caminho onde se encontram os fontes>' -d '<caminho onde se encontra o diretório *matcluster* no aplicativo>' executeSegmenta

4) Edite o arquivo ClusterAnalysis.java
    a. Insira uma entrada para a chamada do seu método na variável 'hashMetodos'.
        i. Primeiro parâmetro: caminho definido no método createNodes.
        ii. Segundo parâmetro: nome do método que irá executar o novo método criado no arquivo Cluster.java.
    b. Crie um novo método público com o nome definido no passo anterior.
    c. Dentro do novo método, siga o mesmo padrão adotado pelos demais métodos do Matlab®.

4.2.1.2. Métodos no R®

Assim como no Matlab®, para criar métodos no R® é necessário também conhecer um pouco da linguagem de programação Java. Siga os seguintes passos para a criação de novos métodos:

1) Crie o novo método no R® de modo a retornar um objeto com pelo menos um componente chamado 'clustering' que será uma matriz n x 1 contendo o índice do agrupamento a que pertence.
2) Edite o arquivo ClusterAnalysis.java
    d. Insira uma entrada para a chamada do seu método na variável 'hashMetodos'.
        i. Primeiro parâmetro: caminho definido no método createNodes.
        ii. Segundo parâmetro: nome do método que irá executar o novo método criado no arquivo Cluster.java.
    e. Crie um novo método público com o nome definido no passo anterior.

f. Dentro do novo método, siga o mesmo padrão adotado pelos demais métodos do R®.

4.2.2. Métodos Internos

Para criar métodos internos é necessário conhecer a linguagem de programação Java. Siga os seguintes passos para a criação de novos métodos:

1) Edite o arquivo Cluster.java
    a. Crie o método de agrupamento de dados como público contendo os parâmetros de entrada necessários.
    b. No final da execução do método, tenha certeza de que a matriz de pertinências 'pertinencias' está atualizada com o resultado da execução do método, e execute, se necessário, o método setCluster() para atualizar o vetor 'cluster' com os índices dos dados de cada agrupamento.
    c. Utilize como base os métodos kmeans e fcm para a construção do novo método.
2) Edite o arquivo ClusterAnalysis.java
    a. Insira uma entrada para a chamada do seu método na variável 'hashMetodos'.
        i. Primeiro parâmetro: caminho definido no método createNodes
        ii. Segundo parâmetro: nome do método que irá executar o novo método criado no arquivo Cluster.java.
    b. Crie um novo método público com o nome definido no passo anterior.
    c. Dentro do novo método, execute o novo método criado no arquivo Cluster.java.
    d. Edite o método createNodes e adicione o nome do novo método criado no lugar apropriado.
    e. Utilize como base os métodos internos kmeans e fcm como base para a construção do novo método.