CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA

CELSO SUCKOW DA FONSECA – CEFET/RJ

Ferramenta web para visualização de dados estelares

Hugo Pinheiro de Aguiar

Orientadores: Eduardo Bezerra, D.Sc.

Maysa Macedo, D.Sc.

Rio de Janeiro

Março de 2014

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA

CELSO SUCKOW DA FONSECA – CEFET/RJ

Ferramenta web para visualização de dados estelares

Hugo Pinheiro de Aguiar

Projeto final apresentado em cumprimento às normas do Departamento de Educação Superior do CEFET/RJ, como parte dos requisitos para obtenção de Grau de Tecnólogo em Sistemas para Internet.

Orientadores: Eduardo Bezerra, D.Sc.

Maysa Macedo, D.Sc.

Rio de Janeiro

Março de 2014

Resumo

De maneira geral, estrelas não nascem isoladamente. Elas comumente têm sua origem em estruturas denominadas aglomerados estelares. As estrelas que compõem esses aglomerados encontram-se aproximadamente a mesma distância e também possuem aproximadamente a mesma idade. Sendo assim, estudar esses aglomerados pode ajudar a responder várias questões sobre a origem do Universo, bem como a história e o destino do nosso próprio sistema solar. No entanto toda a análise passível de ser feita baseia-se unicamente na luz emitida por estes objetos estelares e que pode ser dividida em várias bandas dos espectros visível e invisível ao olho humano. Nesse contexto, o uso de ferramentas computacionais para auxiliar estes estudos não significa apenas uma opção, mas uma necessidade devido à complexidade em analisar o volume de dados coletado. Este trabalho apresenta uma ferramenta WEB de visualização interativa destes dados em 2D e 3D. O objetivo da ferramenta é facilitar a análise exploratória desses dados por meio da geração de diversas perspectivas gráficas dos mesmos.

Abstract

Generally stars are not born alone. They usually have their origin in structures called star clusters. Due to the fact that the stars that compose them are relatively at the same distance and also have about the same age, studying them can help answer many questions about the origins of the universe, the history and fate of our own solar system. However, the whole subject analysis that can be made is based solely on the light emitted by these stellar objects and can be divided into several bands of visible and invisible for the human eye spectrum. In this context, the use of computational tools to assist these studies does not mean just an option but a necessity due to the complexity of analyzing the volume of collected data. Thus, this work presents an interactive visualization WEB tool of these data in 2D and 3D.

Agradecimentos

Agradeço ao astrônomo Alberto Krone-Martins da Universidade de Lisboa-Portugal, que produziu os arquivos de dados sintéticos usados como testes do sistema. Agradeço também aos professores do CEFET-RJ que me ajudaram e mantiveram minha curiosidade e interesse na área e cujos *insights* foram muito valiosos ao longo do curso.

Sumário

1 Introdução 1
1.1 Justificativa 2
1.2 Objetivos 2
1.3 Metodologia 2
1.4 Trabalhos Relacionados 3
1.5 Organização do Trabalho 5
2 Fundamentos da Análise de Dados em Astronomia 6
2.1 Aglomerados Estelares 6
2.2 Magnitudes 7
2.3 Lei do deslocamento de Wien e de Stefan-Boltzmann 8
2.4 Classificação Estelar 10
2.5 Índices de cor 11
2.6 Diagrama baseado em fotometria 11
3 Desenvolvimento do Sistema 13
3.1 Requisitos do Sistema 13
3.2 Arquitetura do Sistema 13
3.3 Modelo de Casos de Uso 14
3.4 Modelo de Classes 16
3.4.1 Exibir Mosaico e Gerar Gráfico 3D 18
3.4.2 Importar e Exportar Dados 21
3.4.3 Mapeamento do VOTable 22
4 Aspectos de Implementação 24
4.1 Arquivos de Dados 24
4.1.1 Arquivo texto DAT 24
4.1.2 Arquivos FITS 26
4.1.3 Arquivo VOTable 27
4.2 Construção da camada de apresentação 30
4.3 Geração das Visualizações 31
4.3.1 Geração de imagem 2D de estrelas 31
4.3.2 Geração dos Diagramas Cor-Magnitude 32

4.3.3 Comunicação ente *Applet* e *Servlet* 32
4.3.4 Representação dos dados em três dimensões 33
4.3.5 Funcionalidade Mosaico 33
5 Conclusões 35
5.1 Considerações Finais 35
5.2 Trabalhos Futuros 35
Referências 36
Apêndice A - Manual do Usuário 38

LISTA DE FIGURAS

Figura 1-1 - Tela do software IRAF.Fonte:(IMAGE REDUCTION AND ANALYSIS FACILITY, 2013) 4

Figura 1-2 - Tela do software TOPCAT. Fonte: (TAYLOR, 2013) 5

Figura 2-1 - Aglomerados. a)Plêiades b)Omega Centauri. Fonte:(Galaxy Photography, 2014). 6

Figura 2-2 - Lei do deslocamento de Wien. Fonte: (ASTRONOMY ONLINE, 2013) 9

Figura 2-3 - Diagrama CMD gerado a partir de dados sintéticos (cortesia de Alberto Krone-Martins). 12

Figura 3-1 - Arquitetura Básica do Sistema 14

Figura 3-2 - Diagrama de Casos de Uso 15

Figura 3-3 - Diagrama de Classes (contendo atributos e operações). 17

Figura 3-4 - Classes em Exibir Mosaico e Gerar Gráfico 3D 18

Figura 3-5 - Classes Presentes em Importar e Exportar Dados 21

Figura 3-6 - Classes presentes no mapeamento de VOTable 23

Figura 4-1 - Arquivo de Dados Sintéticos 25

Figura 4-2 - Tela de entrada da aplicação criada com o Bootstrap. 30

Figura 4-3 - Imagem 2D de estrelas gerada pelo sistema 31

Figura 4-4 - Diagrama CMD gerado pelo sistema 32

Figura 4-5 - Gráfico em três dimensões apresentando as coordenadas x, y e a magnitude visual V. 33

Figura 4-6 - Funcionalidade Mosaico 34

Figura A- 1 - Tela principal do sistema. 38

Figura A- 2 - Tela com as funcionalidades de visualização. 39

Figura A- 3 - Tela de projeção 2D das estrelas com dois diagramas CMD. 40

Figura A- 4 - Tela de geração de imagens personalizadas de acordo com os parâmetros selecionados. 41

Figura A- 5 - Tela que mostra a funcionalidade mosaico e a seleção de um subconjunto sendo refletida nos demais diagramas com a cor amarela e na imagem 2D em azul. 42

Figura A- 6 - Tela que mostra um cubo com pontos que representam as posições X e Y associadas a magnitude aparente. 43

Figura A- 7 - Imagem 2D que representa o brilho de cada estrela de um conjunto de dados, utilizando recursos do HTML5. 44

LISTA DE EQUAÇÕES

Equação 1- Relação entre magnitudes absoluta e aparente 7
Equação 2 - Expressão da Lei de Wien 8
Equação 3 - Lei de Stefan-Boltzmann 9

LISTA DE TABELAS

Tabela 1 - Cores de Calibração de Exemplo e Temperatura Efetiva(K): Fonte:(ZOMBECK, 1990) 11

LISTA DE ABREVIATURAS E SIGLAS

IRAF - Image Reduction and Analysis Facility

TOPCAT - Tool for OPerations on Catalogues and Tables

NOAO - National Optical Astronomy Observatories

J2EE - Java2 Platform Enterprise Edition

JPEG - Joint Photographic Experts Group

CMD - Color Magnitude Diagram

XML - Extensible Markup Language

XSLT - eXtensible Style Language Transformation

CSS - Cascade style sheets

JSON - Javascript Object Notation

FITS - Flexible Image Transport System

NGC - New Generic Catalogue

M45 - Aglomerado de Plêiades na classificação de Messier

NGC5239 - Aglomerado de Omega Centauri na classificação NGC

IVTOA - International Virtual Observatory Alliance

VOTABLE - Virtual Observatory Table

1 Introdução

Aglomerados estelares são estruturas onde ocorre intensa formação de estrelas. Podem ser classificados como abertos (open clusters) contendo centenas de estrelas ou como globulares (globular clusters) com milhares ou mesmo milhões de estrelas. Para estudá-los emprega-se a Astrofotometria que é o segmento da Astronomia responsável pela medição da intensidade luminosa dos astros.

Um problema relevante nesta área de estudo é o de identificar as estrelas pertencentes a determinados aglomerados estelares a partir de dados espectrais gerados de uma imagem de campo. Uma imagem de campo é aquela obtida por meio de telescópio apontado uma determinada região do céu. Uma imagem de campo tipicamente apresenta diversos pontos de maior luminosidade, cada um dos quais correspondendo a um objeto estelar.

Pelo fato de essas imagens terem duas dimensões, estrelas que se situam em uma posição relativa anterior ou posterior ao aglomerado (na linha de visão do telescópio utilizado) são indistinguíveis (na imagem) de estrelas que pertençam a um aglomerado estela. Por conta disso, além da informação de posição do objeto estelar na imagem, outras informações são usadas com a finalidade de distinguir que estrelas efetivamente pertencem a um aglomerado e que outras não pertencem. Essas outras informações correspondem a dados fotométricos do espectro infravermelho.

É fato conhecido na Astronomia que estrelas que tem a mesma origem compartilham características fotométricas similares. Sendo assim, essas características são usadas para a determinação de quais estrelas de uma imagem de campo são pertencentes ao aglomerado estelar em estudo.

Neste trabalho apresentamos uma ferramenta WEB interativa para apoiar a análise destes dados correspondentes a objetos estelares de uma região do céu. O propósito dessa ferramenta é facilitar a análise exploratória de conjuntos de dados posicionais e fotométricos com o propósito de facilitar a diferenciação de estrelas pertencentes a um aglomerado por meio da visualização desses dados em diferentes perspectivas. A ferramenta possui código aberto e foi desenvolvida na linguagem Java. A versão mais atual o sistema desenvolvido encontra-se hospedada e pode ser acessada por meio da seguinte URL: http://teammobile.kinghost.net/StellarCluster.

1.1 Justificativa

Uma ferramenta interativa de análise exploratória pode apoiar o trabalho de físicos e astrônomos no sentido de oferecer uma melhor visualização dos dados estelares gerados por um telescópio. A possibilidade de visualizar dados de diferentes formas pode auxiliar na análise da posição das estrelas que não são constituintes do aglomerado estelar, mas que devido as suas posições relativas na imagem apresentam-se como tais.

O software mais utilizado por astrônomos para análise de dados originados de medições por telescópios até o presente momento é o IRAF. Um complicador no uso do IRAF é que esse sistema está disponível apenas para a plataforma Linux. Soma-se a isso o fato de que a interface do IRAF, que é via linha de comando, dificulta sua utilização. Nesse contexto, uma ferramenta na WEB, como a que propomos neste trabalho, fornece alta disponibilidade e dispensa o processo de instalação.

1.2 Objetivos

O objetivo principal a ser atingido é construir uma ferramenta WEB que permita a análise visual de dados estelares. Essa ferramenta poderá ser utilizada por astrônomos e estudantes da área de Astronomia com o intuito de apoiar uma análise visual das características das estrelas localizadas em uma região do céu. Esperamos que as visualizações gráficas desses dados oferecidas por nossa ferramenta podem fornecer um entendimento melhor sobre a natureza dos dados estudados.

Como objetivo secundário deste trabalho deseja-se apresentar a integração de diferentes tecnologias de visualização gráfica e de linguagem de desenvolvimento WEB, mostrando que é possível criar plataformas avançadas de visualização no ambiente WEB.

1.3 Metodologia

A estrutura da ferramenta WEB será constituída basicamente de uma parte J2EE Servlet responsável pela manipulação dos arquivos de dados e de uma parte desenvolvida como Applet, responsável pela exibição e manipulação destes dados na forma de gráficos. Para facilitar a manipulação gráfica dos dados na Applet será utilizada uma API Open Source disponibilizada pelo Processing que trabalha utilizando recursos do OpenGL possuindo também, várias funções de manipulação de estruturas de dados.

O Processing é uma linguagem de programação e também um ambiente de desenvolvimento de código livre. Desde 2001, passou a ser utilizada para promover o ensino de software dentro das artes. Inicialmente foi criado para servir como um recurso de software e para ensinar os fundamentos da programação de computadores em um contexto visual. Com o passar do tempo, evoluiu para uma ferramenta de desenvolvimento profissional. Atualmente, há uma comunidade online ativa composta por milhares de estudantes, artistas, designers, pesquisadores e entusiastas que utilizam processamento de aprendizagem, desenvolvimento de protótipos e produção. É bem documentada, com vários exemplos disponíveis.

Neste trabalho de conclusão de curso não será utilizado o ambiente de programação do Processing, mas sim uma API que é disponibilizada na forma de uma biblioteca jar. O Processing também dispõe de uma versão Javascript+HTML5 que constitui uma alternativa ao uso de Java Applets, porém existem restrições quanto às combinações de browser e placas de vídeo suportadas.

A base deste módulo da ferramenta é desenhar gráficos CMD de forma dinâmica a partir de dados simulados de aglomerados estelares. Estes gráficos são desenhados tendo-se como eixo Y a magnitude visual ou relativa V e como eixo X um dos índices de cor obtidos a partir das outras colunas do dado de entrada. Segundo (CLAUDIA.RODRIGUES, 2011), um índice de cor representa certo nível de radiação corrigida e serve para representar principalmente a temperatura de um objeto estelar através da lei de Wien como descrito por meio do – vídeo de Bob Eagle (Star Classification - A Level Physics, 2013). Mais adiante na Seção 2.5, veremos que existe uma relação entre o nível de radiação e a temperatura efetiva do objeto estelar.

1.4 Trabalhos Relacionados

A ferramenta IRAF, acrônimo para *Image Reduction and Analysis Facility*, é um sistema de uso geral para análise de dados astronômicos(IMAGE REDUCTION AND ANALYSIS FACILITY, 2013). É um projeto desenvolvido pelo *National Optical Astronomy Observatories* (NOAO). Este software está disponível atualmente para os sistemas operacionais Unix (Linux) e Mac OS. Não há até o presente momento uma distribuição Windows desta ferramenta. Uma das telas do software é apresentada na Figura 1-1.

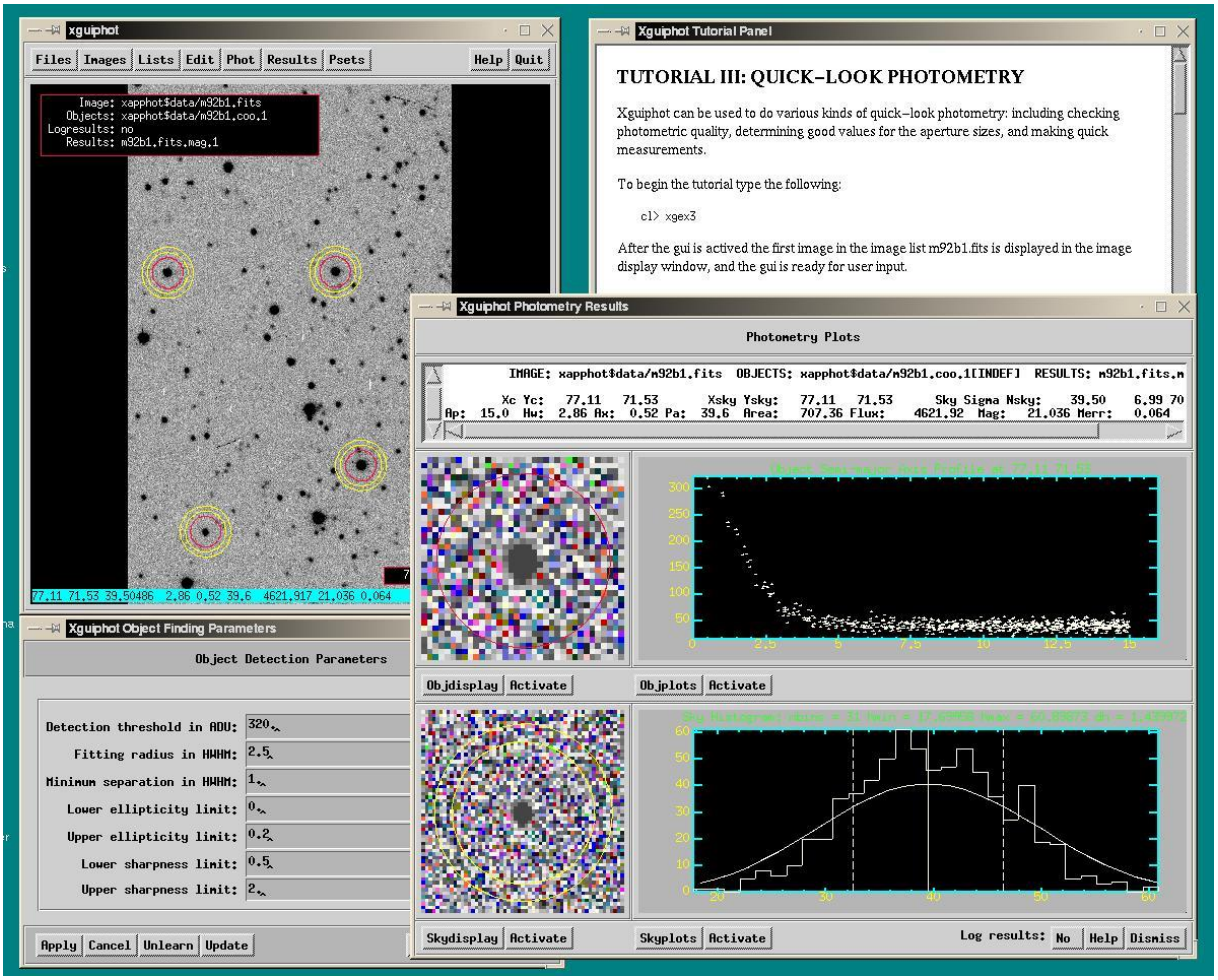


Figura 1-1 - Tela do software IRAF. Fonte:(IMAGE REDUCTION AND ANALYSIS FACILITY, 2013)

Outra ferramenta conhecida no meio acadêmico é a TOPCAT, acrônimo para *Tool for Operations on Catalogues And Tables* (TAYLOR, 2013). Essa ferramenta é ao mesmo tempo um visualizador gráfico interativo e um editor de dados tabulares. Seu objetivo é fornecer a maior parte das ferramentas que os astrônomos precisam para analisar e manipular catálogos de origem e outras tabelas. Entretanto, esta ferramenta pode ser usada também para fontes de dados não-astronômicas. Compreende uma série de formatos diferentes astronomicamente importantes (incluindo os formatos FITS, VOTable e CDF), mais formatos podem ser adicionados e oferece uma variedade de maneiras de ver e analisar as tabelas, incluindo um navegador para os dados de células, informações sobre a tabela e metadados de coluna, cálculo de estatísticas e associação de tabelas. Usando uma linguagem de expressão baseada em Java, novas colunas podem ser definidas e subconjuntos de linhas selecionadas para a análise em separado. Os dados da tabela e metadados podem ser editados e a tabela modificada resultante pode ser escrita em uma ampla gama de formatos de saída como mostra a Figura 1-2.

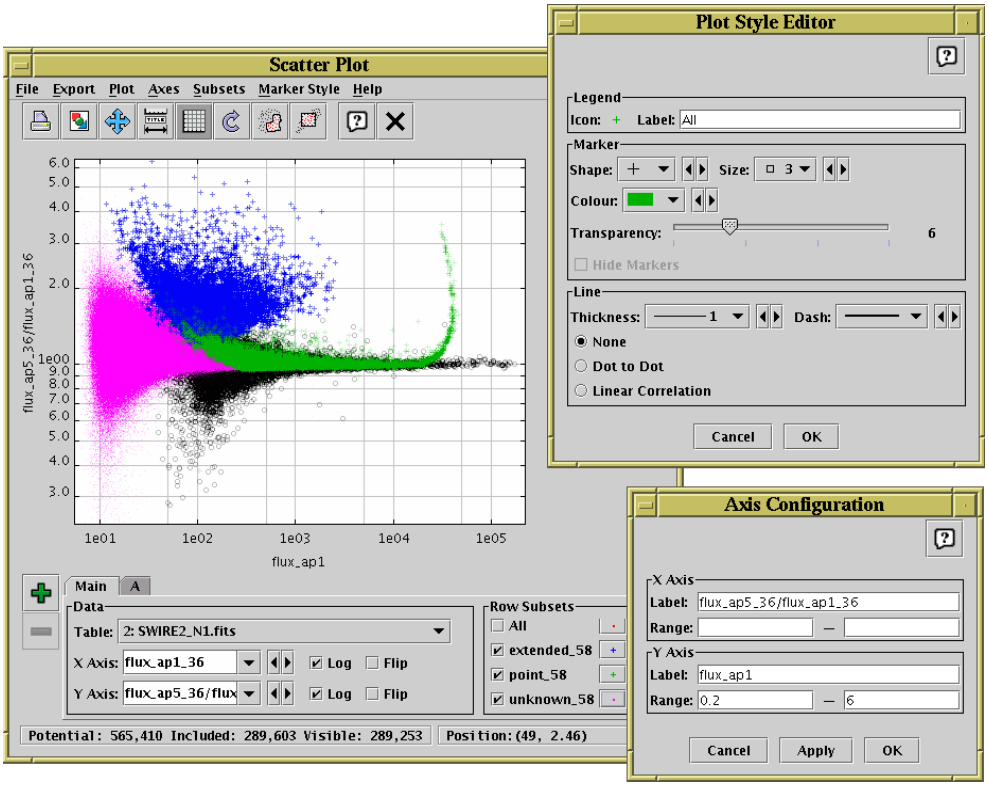


Figura 1-2 - Tela do software TOPCAT. Fonte: (TAYLOR, 2013)

Projetos elaborados utilizando Processing estão disponíveis na página principal do projeto. A API tem uma documentação bastante abrangente e com vários exemplos sendo que alguns destes fazem uso da biblioteca Javascript+HTML5 *ProcessingJs*. A lista de projetos é atualizada mensalmente.

1.5 Organização do Trabalho

Para melhor compreender os objetivos deste trabalho, é necessário detalhar a conceitos relacionados aos aglomerados estelares e aos diagramas utilizados bem como suas propriedades. Este detalhamento é feito no Capítulo 2. No Capítulo 3, são apresentados os requisitos e os artefatos de modelagem da ferramenta que propomos. Aspectos relacionados à construção da aplicação propriamente dita tais como geração dos gráficos 2D e 3D, a importação e exportação dos arquivos de dados, estarão relacionados no Capítulo 4. No Capítulo 5 apresentamos as considerações finais e a descrição de eventuais trabalhos futuros. Foi produzido um manual do sistema para o usuário final que está apresentado no Apêndice A.

2 Fundamentos da Análise de Dados em Astronomia

Neste Capítulo são apresentados os conceitos relacionados à Astronomia necessários para o entendimento do problema e da solução proposta. O objetivo é apresentar a terminologia e conceitos utilizados na análise de dados em Astronomia bem como caracterizar os tipos de dados que serão processados pelo sistema.

2.1 Aglomerados Estelares

Aglomerados estelares, já mencionados anteriormente, são um sistema contendo centenas, milhares ou mesmo centenas de milhares de estrelas de diferentes magnitudes unidas por um campo gravitacional comum (DEPARTAMENTO DE ASTRONOMIA DA UFRGS, 2014). De acordo com Ridpath (RIDPATH, 2007, p. 70), os aglomerados podem ser classificados em dois tipos: abertos e globulares.

Um aglomerado aberto contém centenas de estrelas. Um exemplo é o aglomerado de Plêiades (denominado também como M45 na classificação de Messier) ilustrado na Figura 2-1a.

Já os aglomerados globulares caracterizam-se por conter milhares ou mesmo milhões de estrelas. Um exemplo é o aglomerado Omega Centauri (denominado também como NGC5239 na classificação NGC - *New General Catalogue*), ilustrado na Figura 2-1b.

Figura 2-1 - Aglomerados. a)Plêiades b)Omega Centauri. Fonte:(Galaxy Photography, 2014).

2.2 Magnitudes

Conforme definição de Ridpath (RIDPATH, 2007, p. 135), no domínio estelar, magnitude é basicamente uma medida do brilho de um objeto estelar. Porem há basicamente dois tipos de mensuração desta grandeza: Magnitude Aparente e Magnitude Absoluta.

Magnitude Aparente é uma escala de comparação entre o brilho de estrelas do ponto de vista do observador. Sua criação é atribuída a Hiparco (astrônomo greco-otomano) há cerca de 2000 anos (FORBES, 2009). Inicialmente era compreendida entre 1 a 6, sendo o valor 1 referente a estrela mais brilhante visível e 6 correspondente a de menor visibilidade. Posteriormente com o advento de telescópios esta escala foi estendida para contemplar objetos além do alcance da visão humana e também objetos próximos como o Sol e a Lua. No século XVIII foi formalizada como uma escala logarítmica onde se definiu que a razão ente a magnitude 1 e 6 seria igual a 100. Isto equivale a dizer que cada grau de magnitude corresponde a um aumento de aproximadamente 2,5 vezes.

A título de exemplo, o Sol possui magnitude aparente de -26,74, a Lua cheia tem magnitude de -12,6 e a estrela mais brilhante, Sirius possui -1,45 de magnitude aparente.

Magnitude Absoluta é definida como a magnitude aparente que o objeto teria se o mesmo estivesse a uma distância de 10 parsecs (aproximadamente 32,6 anos luz). Magnitudes Absolutas geralmente variam entre -10 (mais brilhantes) e +17 (mais fracas). Note que para determinar a magnitude absoluta é necessário conhecer a distância até o objeto. Como exemplos, citam-se o Sol que possui Magnitude Absoluta de +4,83 e a estrela Sirius possui +1,7 dessa magnitude.

A relação entre magnitude absoluta e aparente é estabelecida na Equação 1:

Equação 1

$$M = m - 5((\log_{10} D) - 1)$$

em que:

M é a Magnitude Absoluta,

m é a Magnitude Aparente e

D representa a distância em Parsecs até o objeto.

Parsec (COSMOS, 2014) é uma unidade utilizada usada em trabalhos científicos de astronomia para representar distâncias estelares. Surgiu para representar a unidade de

distância das estrelas calculada por paralaxe (diferença das distâncias de dois pontos de observação diferentes a um mesmo objeto alvo). A palavra parsec surgiu da contração das palavras *paralax* (paralaxe) e *second* (segundo). A definição é derivada de uma razão trigonométrica entre uma unidade astronômica (distância entre terra e o sol) e a distância de uma estrela. Para um parsec, o ângulo vale um segundo de arco de deslocamento. Corresponde a 3,26 anos-luz. O principal motivo para sua utilização é o fato de indicar distâncias medidas diretamente através da observação astronômica de paralaxes sem a necessidade de realizar conversões.

2.3 Lei do deslocamento de Wien e de Stefan-Boltzmann

De acordo com a lei de Wien, é possível determinar a temperatura de um corpo negro baseando-se apenas em sua cor. Um corpo negro é definido como sendo um objeto opaco e não reflexivo (ASTRONOMY ONLINE, 2013). Esta lei determina que o comprimento de onda da luz emitida pelo corpo negro é inversamente proporcional a sua temperatura. Isto significa que quanto mais azulado for um objeto estelar, maior sua temperatura ao passo que quanto mais avermelhado for um objeto mais frio ele será. Isto pode parecer contra intuitivo uma vez que temos a tendência de associar a cor vermelha ao calor. Para efeito de comparação, coloração amarela do Sol corresponde a uma temperatura superficial da ordem de 5000 graus Kelvin.

A Lei de Wien pode ser expressa pela Equação 2:

Equação 2

$$\lambda_{\max} = \frac{b}{T}$$

em que:

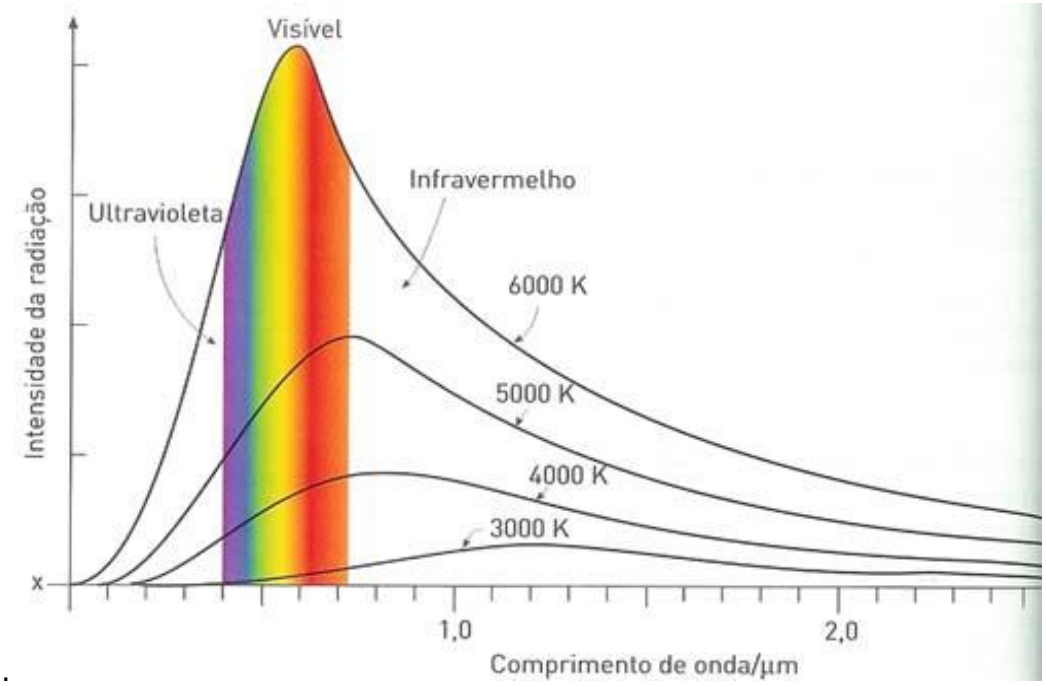
$\lambda_{\max}$ é o comprimento de onda (em metros) referente à intensidade da radiação eletromagnética máxima;

T é a temperatura do corpo negro em graus kelvin (K)

b é uma constante de proporcionalidade chamada constante de dispersão de Wien, cujo valor é 2.8977685×10^{-3} kelvins-metro.

Esta relação pode ser mais bem compreendida através da Figura 2-2 na qual fica evidenciado que o comprimento correspondente à intensidade máxima de radiação varia conforme a temperatura.

A Lei de Stefan-Boltzmann declara que a potência irradiada por unidade de área varia apenas com a temperatura, ela não depende do material e de outras características do corpo. Corpos reais irradiam em menos quantidade de energia por unidade de área em comparação com o corpo negro. Para calcular a energia irradiada por esses corpos é necessária a inclusão de um parâmetro denominado emissividade e, a emissividade depende das características do material (cor, composição de sua superfície), seu valor fica entre zero e um.



.

Figura 2-2 - Lei do deslocamento de Wien. Fonte: (ASTRONOMY ONLINE, 2013)

R (energia irradiada por segundo) de um corpo negro é diretamente proporcional à sua temperatura *T* elevada à quarta potência e também diretamente proporcional à área *A* da superfície emissora como mostrado na Equação **3**.

Equação 3

$$R = \sigma T^4 * A$$

em que $\sigma = 5{,}67 \times 10^{-8} W/m^2K^4$ é a constante de Stefan-Boltzmann.

Em conjunto, estas duas leis permitem calcular a temperatura, luminosidade e volume de uma estrela.

2.4 Classificação Estelar

Estrelas são classificadas conforme a magnitude absoluta e temperatura ou volume e luminosidade (TODD, 1922). Com relação ao sistema Morgan–Keenan as estrelas são classificadas de acordo com a temperatura usando as seguintes letras O, B, A, F, G, K e M, em que O representa alta temperatura e sequencialmente M representa a mais baixa temperatura (MORGAN, 1943).

A razão para este arranjo singular de letras é histórica segundo a AMATEUR ASTRONOMERS, INC(AMATEUR ASTRONOMERS, INC, 2013).A classificação anterior do espectro realizada por Angelo Secchi dividiu estrelas em4gruposcom linhas de destaque da série do hidrogênio: grupo I, com um subtipo que representa muitas das estrelas na constelação de Orion, grupo II aquelas com espectros que, como o Sol, mostrou cálcio e as linhas de sódio; grupo III estrelas coloridas cujos espectros mostraram bandas largas e grupo IV estrelas de carbono. Em meados de 1880, o astrônomo Edward C. Pickering, ao fazer um levantamento dos espectros estelares usando o método objetivo de prisma, um primeiro resultado desse trabalho foi o Catálogo de Draper Stellar Spectra, publicado em 1890, tendo Williamina Fleming, classificado a maior parte do espectro neste catálogo. É utilizado um esquema em que as classes Secchi usadas anteriormente (I a IV) foram divididas em classes mais específicas, dadas letras de A até N. Além disso, foram utilizadas as letras O, P e Q, O para estrelas cujos espectros consistiam principalmente de linhas brilhantes, P para nebulosas planetárias, e Q para as estrelas não que não podem se encaixar em qualquer outra classe. Esta classificação ainda subdivide-se em um intervalo que varia de 0 a 9 entre uma classe e outra adjacente. Por exemplo, a classe G subdivide-se em G0 a G9. Por último, segundo a luminosidade ou volume pode ser classificada como I, II, III, IV e V (em romanos) em ordem decrescente. Sendo assim o Sol é uma estrela classificada como G2V, ou seja, situa-se a dois décimos entre as classes G0 e K0 e possuindo volume da ordem de grandeza V. Rigel, que é estrela azulada, possui a classificação B8I.

2.5 Índices de cor

É uma expressão numérica que determina a cor “real” de um objeto. Os índices de cor são utilizados para determinar a temperatura de um objeto estelar de acordo com a Lei de Wien para radiação de corpos escuros (SAN DIEGO STATE UNIVERSITY, 2013). Para estrelas, estes índices são fatores determinantes na classificação estelar. Como exemplo, o Sol possui um índice B-V de 0.656 (± 5000 K) e a estrela Rigel –0.03 (± 12000 K). Ainda pode-se afirmar que quanto mais negativo for o valor neste índice maior a temperatura efetiva destes objetos e, portanto mais azulados. A Tabela 1 contém um exemplo de relação das classes com seus respectivos índices de cor.

Tabela 1 - Cores de Calibração de Exemplo e Temperatura Efetiva(K): Fonte:(ZOMBECK, 1990)

Classe	B–V	U–B	V–R	R–I	Teff (K)
O5V	–0.33	–1.19	–0.15	–0.32	42,000
B0V	–0.30	–1.08	–0.13	–0.29	30,000
A0V	–0.02	–0.02	0.02	–0.02	9,790
F0V	0.30	0.03	0.30	0.17	7,300
G0V	0.58	0.06	0.50	0.31	5,940
K0V	0.81	0.45	0.64	0.42	5,150
M0V	1.40	1.22	1.28	0.91	3,840

2.6 Diagrama baseado em fotometria

Determinar a magnitude absoluta de uma estrela requer que seja conhecida também sua distância o que pode ser inviável para um conjunto grande de estrelas como no caso de um aglomerado estelar. O que é feito neste caso é construir diagramas Cor-Magnitude (CMD) como mostra a Figura 2-3, utilizando-se no lugar da magnitude absoluta, a magnitude aparente ou visual, pois esta última é mais simples de ser medida por meio de fotografias obtidas com o uso de filtros padronizados além do fato desta ser independente da distância. Por este motivo, os diagramas gerados pela ferramenta deste presente trabalho serão, em princípio, apenas diagramas CMD´s. Para calcular a temperatura determina-se a cor "real" do objeto através de sistemas de filtros. O mais utilizado nesse caso é o sistema UBV, descrito a seguir.

O sistema fotométrico UBV, também chamado sistema Johnson ou sistema Johnson-Morgan (CLAUDIA.RODRIGUES, 2011) utiliza um conjunto de três filtros especiais, designados pelas letras U, B, e V, que filtram a luz da estrela, deixando passar apenas bandas específicas do espectro eletromagnético. As bandas são posicionadas, na região do ultravioleta (U), na região do azul (B), e na região da luz visível (V) do espectro luminoso. Desta forma, pode-se medir a magnitude da estrela em cada uma das bandas e determinar os índices de cor, B-V e U-B.

Os filtros são construídos de tal maneira que os comprimentos de onda médios da banda passante estejam centrados em 364 nm para a banda U, 442 nm para a banda B, e 540 nm para a banda V. Uma estrela vermelha é mais brilhante (possui magnitude menor) no filtro V do que no filtro B. Do mesmo modo, uma estrela azul possui magnitude menor no filtro B do que no filtro V.A diferença entre essas duas magnitudes, é denominada índice de cor (B-V) e é uma medida da cor da estrela, já que uma estrela vermelha terá um índice de cor (B-V) maior do que o de uma estrela azul. De modo análogo, define-se o índice de cor (U-B).Por definição, os índices de cor, B-V e U-B, são iguais a zero para estrelas de tipo espectral A0V tais como a estrela Vega (ZOMBECK, 1990).

Além dos filtros UBV são usados também os filtros R, I, J, H, K reservados para outras faixas de frequência gerando índices de cor associados às temperaturas mais baixas.

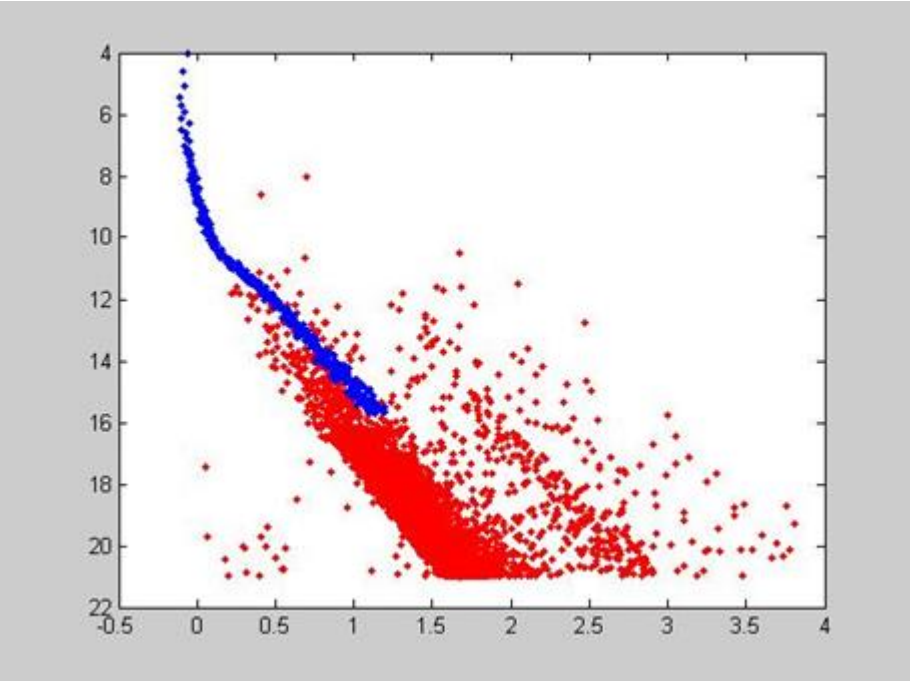


Figura 2-3 - Diagrama CMD gerado a partir de dados sintéticos (cortesia de Alberto Krone-Martins).

3 Desenvolvimento do Sistema

Neste Capítulo apresentamos aspectos relativos ao projeto do sistema como os requisitos principais do sistema e descrições dos diagramas de caso de uso e de classes.

3.1 Requisitos do Sistema

O sistema deve oferecer como requisito principal a visualização dos conjuntos dados na forma de gráficos CMD com uma função interativa de seleção de pontos, além de oferecer uma visualização 3D interativa. Também deve disponibilizar uma funcionalidade de mosaico, em que vários CMD´s são apresentados lado a lado numa disposição de grade sendo que o último gráfico representa uma plotagem das posições x e y e de magnitude aparente de cada objeto no conjunto de dados. Este mosaico é interativo e a seleção de um subconjunto de dados em um dos CMD´s implica no destaque dos mesmos pontos referentes ao subconjunto em questão nos demais CMD´s.

Como requisito não funcional deve contemplar a importação de dados em arquivos de dados como texto puro (DAT) além de suportar os formatos estruturados FITS e VOTable descritos no Capítulo 4.

3.2 Arquitetura do Sistema

A arquitetura básica do sistema consiste de uma parte cliente representada pelo browser de internet do usuário contendo Applets ou páginas com HTML 5 que podem utilizar as APIS Dojo, Bootstrap e ProcessingJs e que se comunicam com o servidor sendo que estas interfaces são implementadas com HTTPServlets. Os dados do usuário são acessados a partir de uma estrutura de cachê como mostra a Figura 3-1.

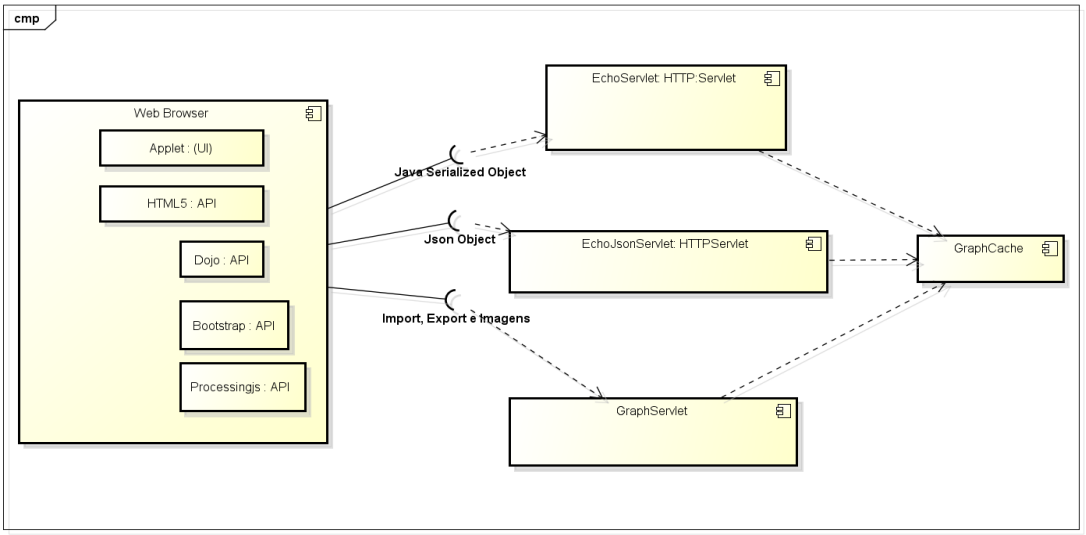


Figura 3-1 - Arquitetura Básica do Sistema

3.3 Modelo de Casos de Uso

A versão do sistema desenvolvido contempla o diagrama de casos de uso apresentado na Figura 3-2. Neste diagrama, o ator representa um possível usuário sem distinção de perfil e dispensando processo de autenticação.

O caso de uso 'Importar dados' representa a funcionalidade que realiza o *upload* do arquivo de dados que é o insumo necessário ao seu posterior processamento. Não possui pré-condições. O usuário seleciona o tipo de arquivo que deseja fornecer como upload. As opções disponíveis são FITS, VoTable e DAT (tipo texto), sendo este último a opção padrão. Caso o arquivo não possua o tipo correto ou encontre-se corrompido, o usuário é informado do não prosseguimento do caso de uso. No caso de importação bem sucedida os dados estarão disponíveis para posterior processamento.

O caso de uso 'Utilizar conjunto Pré-definido de dados' representa uma extensão da funcionalidade do caso de uso 'Importar dados' onde se escolhe trabalhar com um conjunto de dados pré-definidos. Foi elaborado no sentido de facilitar a operação da ferramenta por parte do usuário.

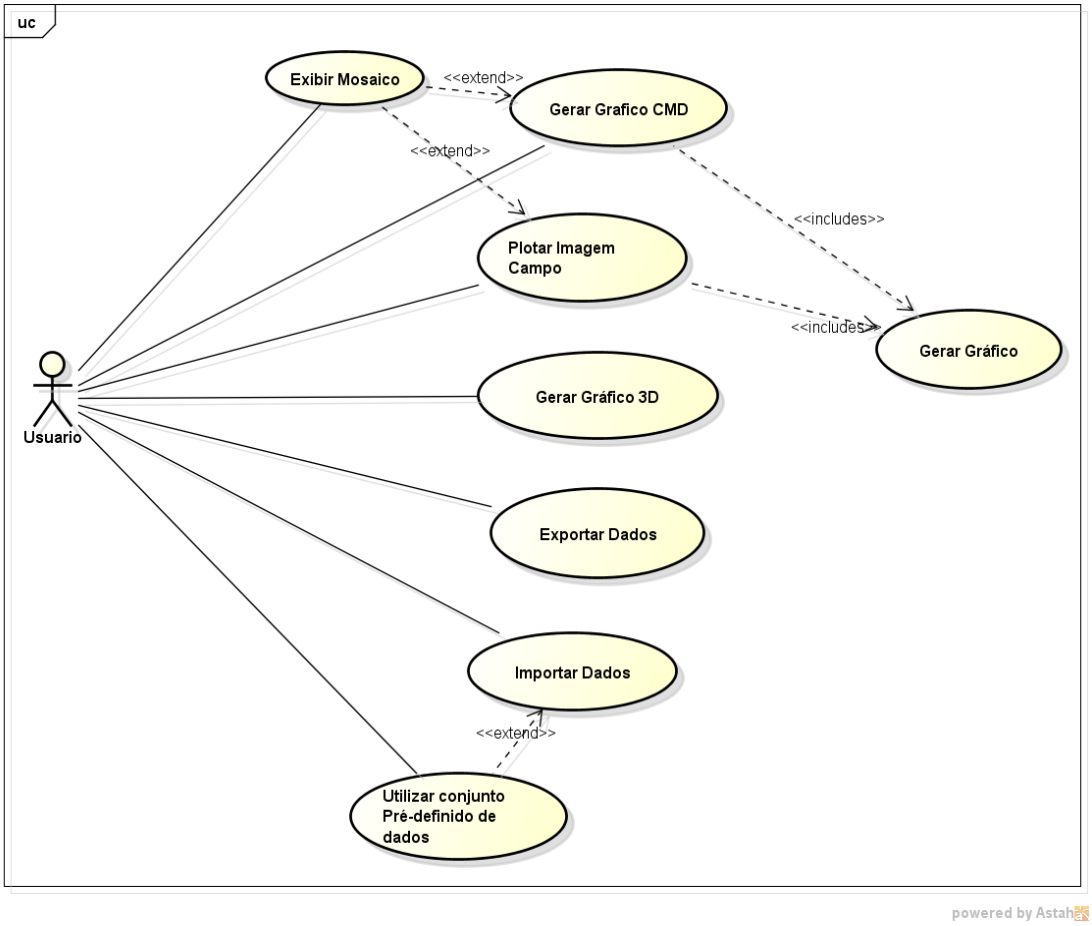


Figura 3-2 - Diagrama de Casos de Uso

O caso de uso 'Plotar imagem de campo' cria uma imagem que se constitui de uma representação gráfica gerada em duas dimensões a partir das informações de posição e magnitude contidas no arquivo de dados previamente carregado a partir de um dos casos de uso anteriores. Como tal, é uma funcionalidade relativamente simples, mas que permite obter uma referência visual do campo tratado.

'Gerar Gráfico CMD' produz gráficos CMD com as informações de magnitudes presentes no arquivo. Um CMD típico contempla uma das colunas do conjunto de dados como eixo Y, sendo por padrão a magnitude visual V.O eixo X pode ser um dos índices de cor ou uma das outras colunas do conjunto de dados. Foi estendido para trabalhar a partir de uma seleção fornecida pelo usuário visualizando-a em cores diferenciadas.

'Exibir Mosaico' é uma extensão de 'Gerar Gráfico CMD' que exibe vários gráficos CMD junto com uma imagem do campo em duas dimensões simultaneamente. Esse mosaico é gerado para o mesmo conjunto de dados desde que exista um número satisfatório de dados. Há também a possibilidade de visualizar uma determinada seleção de objetos nestes gráficos.

'Gerar Gráfico 3D' gera um gráfico em três dimensões. Por padrão, os eixos X e Y correspondem aos homônimos do conjunto de dados e o eixo Z, a magnitude V. Este gráfico é interativo e permite que seja movimentado com o auxílio das setas do teclado.

'Exportar dados' permite que o conjunto de dados corrente possa ser salvo. O usuário seleciona o tipo de arquivo que deseja obter como resultado da exportação e efetua o download. As opções disponíveis são FITS, VoTable e DAT (tipo texto), sendo este último a opção padrão.

3.4 Modelo de Classes

O diagrama de classes referente à implementação dos casos de uso resultou na seguinte versão que está ilustrada na Figura 3-3. A implementação resultou num número relativamente grande de classes e operações e respectivos atributos. Portanto apresentamos diagramas segmentados que não constituem visões de classes participantes por abrangerem mais de um caso de uso.

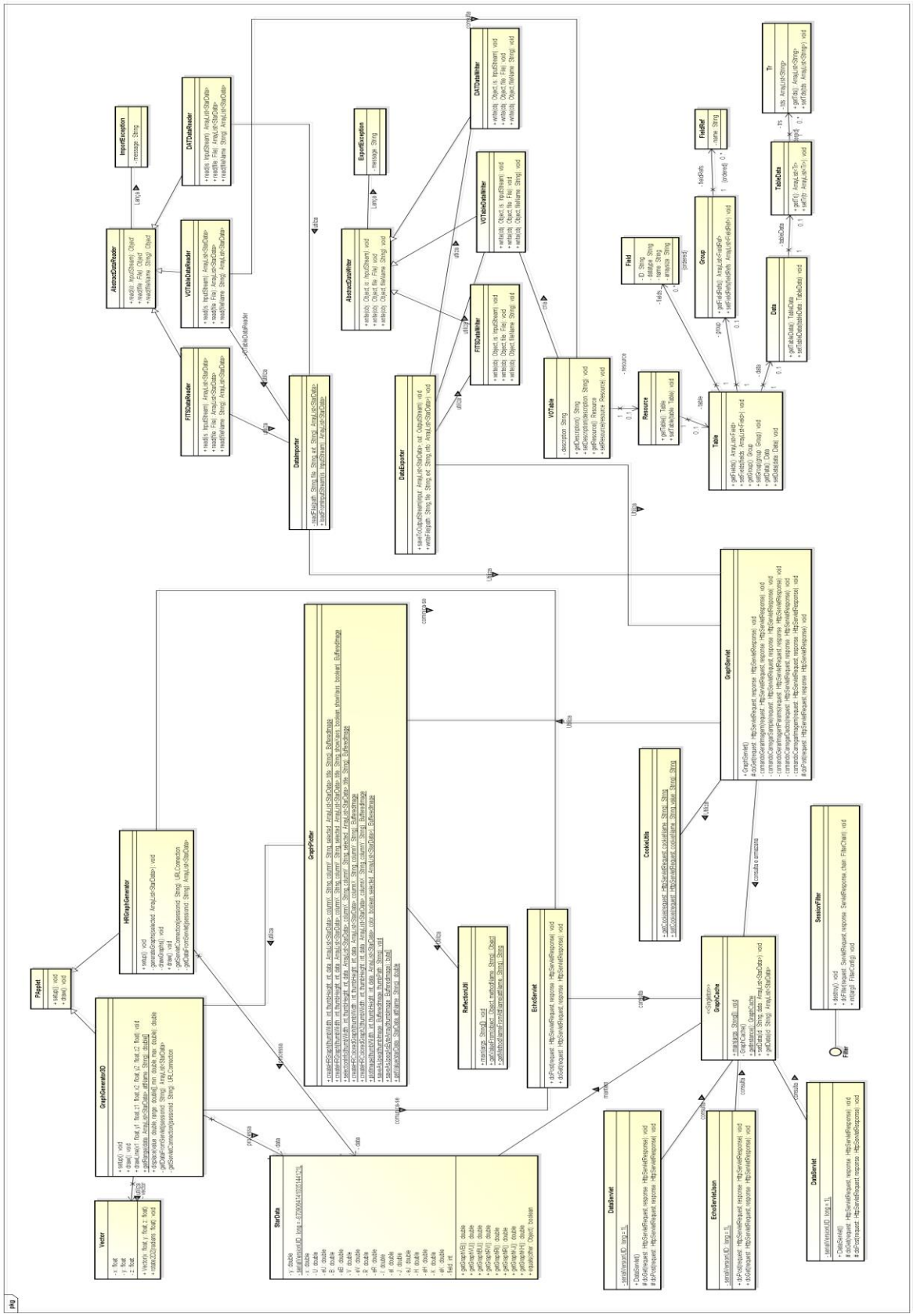


Figura 3-3 - Diagrama de Classes (contendo atributos e operações).

3.4.1 Exibir Mosaico e Gerar Gráfico 3D

Como mostra a Figura 3-4, estamos tratando os dois casos de uso pela similaridade do ponto de vista do funcionamento.

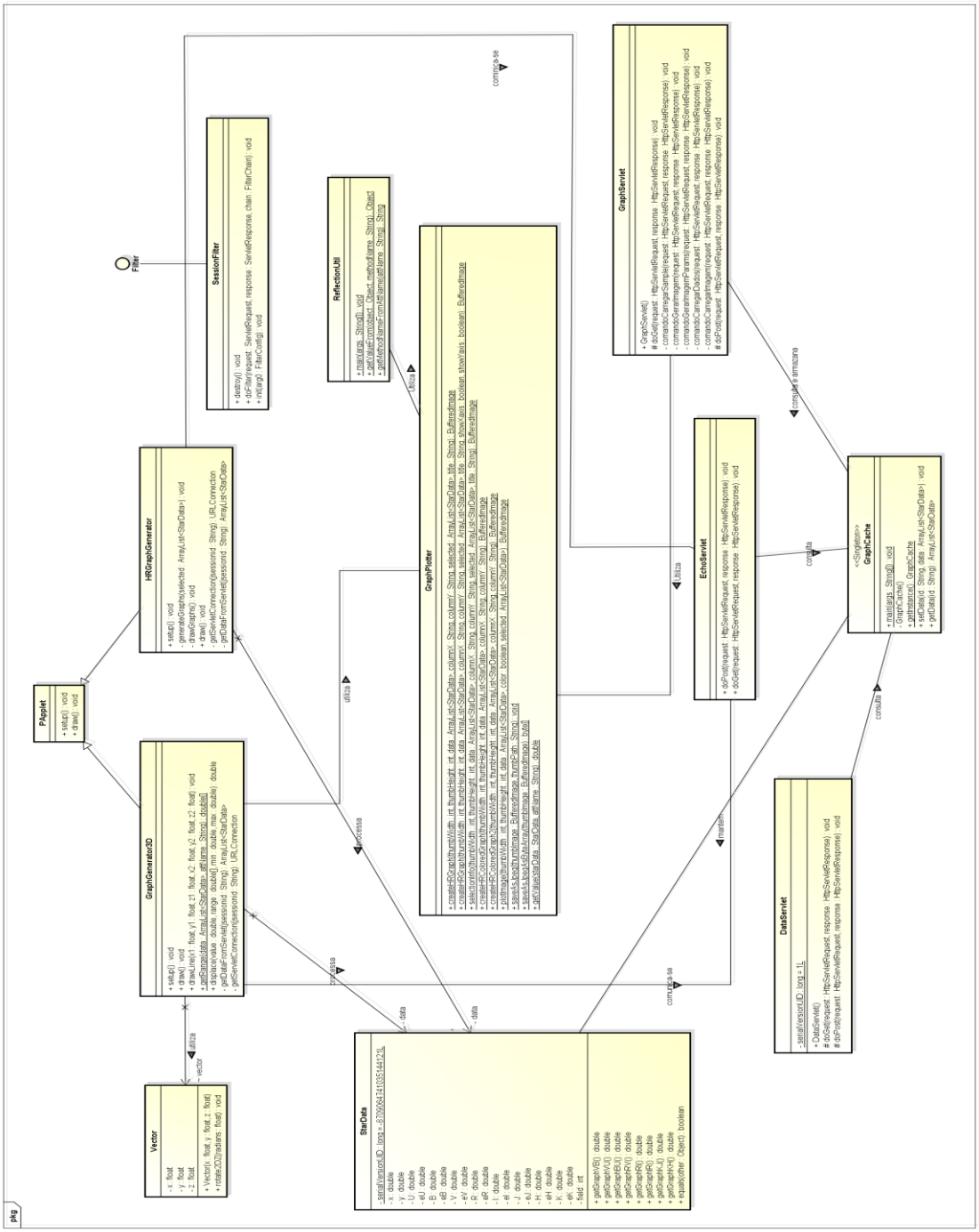


Figura 3-4 - Classes em Exibir Mosaico e Gerar Gráfico 3D

A classe *StarData* é um *ValueObject* que representa uma linha de um arquivo de dados de entrada. É a unidade básica para o processamento dos gráficos que contém as coordenadas x e

y do objeto na imagem assim como as magnitudes relativas e seus erros associados. Foram implementadas algumas funções de conveniência que obtêm os índices de cor relacionados a cada par de magnitudes. São elas *getGraphVB*() (B-V) , *getGraphVU*() (U-V) , *getGraphBU*() (U-B), *getGraphRV*() (V-R), *getGraphRI*() (R-I), *getGraphIR*() (R-I), *getGraphKJ*() (J-K) e *getGraphKH*() (H-K). Todos os métodos retornam o tipo de dado *double*.

HRGraphGenerator é uma subclasse de *PApplet* que é o ponto de extensão do *Processing* resultando em uma Applet contendo uma área de manipulação gráfica. HRGraphGenerator estende a classe *Papplet* do *Processing* cujos métodos principais são *setup()* e *draw()*. Durante a execução do método *setup()* é chamado o método *getDataFromServlet()* que realiza a comunicação entre a *Applet* e o *servletEchoServlet* obtendo assim o conjunto de dados do usuário. Na chamada do método *draw()* é invocado o método *generateGraphs()* que realiza o desenho dos gráficos do mosaico. Pela especificação do *Processing* o método *draw*() pode ser chamado diversas vezes como na ocasião de uma modificação na janela do navegador ou através da interação do mouse.

3DGraphGenerator também é uma subclasse de *PApplet* e é a classe responsável por fazer a manipulação de dados 3D. Assim como na classe HRGraphGenerator, a execução do método *setup()*invoca o método *getDataFromServlet()* que realiza a comunicação entre a *Applet* e o *servletEchoServlet* para obtenção dos dados. Criar uma visualização 3D no *Processing* requer passos a mais do que um desenho em duas dimensões. Torna-se necessário também configurar uma câmera para que o usuário possa interagir com o conjunto de dados visualizado. Optou-se por utilizar as setas do teclado para movimentar esta câmera.

SessionFilter é uma classe que implementa a interface JAVA de *javax.servlet.Filter*. Utilizada para verificar se o usuário possui um conjunto de dados carregado em escopo de sessão. Caso não haja nenhum ele é automaticamente levado para a tela inicial da aplicação. Esta verificação é feita no método *doFilter()* da classe.

GraphServlet é um *HTTPServlet* que possui a função de produzir imagens JPEG a partir de uma coleção de objetos *StarData*. Utiliza a classe auxiliar *GraphPlotter* que tem a função de produzir gráficos. Internamente utiliza a classe *ReflectionUtil* que é uma classe utilitária contendo métodos para obtenção de dados através do uso de recursos da API *Reflection* do JAVA.

GraphPlotter contem vários métodos utilitários para geração de gráficos. O método *createHRGraph(int, int, ArrayList<StarData>, String, String, ArrayList<StarData>, String, boolean, boolean) : BufferedImage* retorna um gráfico CMD baseando-se nas duas dimensões

enviadas como parâmetros que determinam a largura e a altura da imagem a ser gerada respectivamente. É fornecido também um *ArrayList* de objetos *StarData* que representa o conjunto de dados do usuário e os nomes das colunas que compõem os eixos Y e X nesta ordem. Opcionalmente pode-se passar um segundo *ArrayList* de objetos *StarData* que é um subconjunto do conjunto de dados principal como dados selecionados, alem de três outros parâmetros : uma *String* representando o título do gráfico, e duas variáveis do tipo *boolean* indicando se os eixos X e Y devem ou não ser exibidos. O *método plotImage(int, int, ArrayList<StarData>, boolean, ArrayList<StarData>): BufferedImage* produz uma imagem do campo de estrelas 2D. De forma análoga ao método anterior recebe também as dimensões finais e dois *ArrayLists* de objetos *StarData* representando o conjunto de dados e o conjunto de seleção. O método *createHRColoredGraph(int, int, ArrayList<StarData>, String, String): BufferedImage* produz uma versão do CMD com uma variação de cores associada à magnitude V.

DataServlet é um *HTTPServlet* que implementa o envio de arquivos de dados. Baseado no nome do arquivo de dados que é passado como parâmetro para os métodos *get*() e *post*(), o *servlet* envia-o como um texto *(content-type=text/plain).*

EchoServlet é um *HTTPServlet* fornece o conjunto de dados para as applets HRGraphGenerator e 3DGraphGenerator. Este servlet verifica os atributos de sessão e envia o conjunto de dados, um *ArrayList* de objetos *StarData*, como um objeto JAVA serializado *(content-type=application/x-java-serialized-object).*

EchoJsonServlet é um *HTTPServlet* fornece o conjunto de dados para as páginas que utilizam HTML5. Este servlet verifica os atributos de sessão e envia o conjunto de dados do usuário como um objeto Json serializado *(content-type=application/json).*

GraphCache é uma classe que segue o Padrão de projeto *Singleton*. Esta classe contém uma tabela *Hash* que representa o mapeamento entre o identificador da sessão do usuário e o conjunto de dados que foi carregado. Funciona como uma cópia dos dados em sessão e é utilizado pelos servlets *EchoJsonServlet* e *EchoServlet* para enviar os conjuntos de dados correspondentes.

3.4.2 Importar e Exportar Dados

A Figura 3-5 mostra as classes presentes na implementação dos dois casos de Uso (importar e exportar dados).

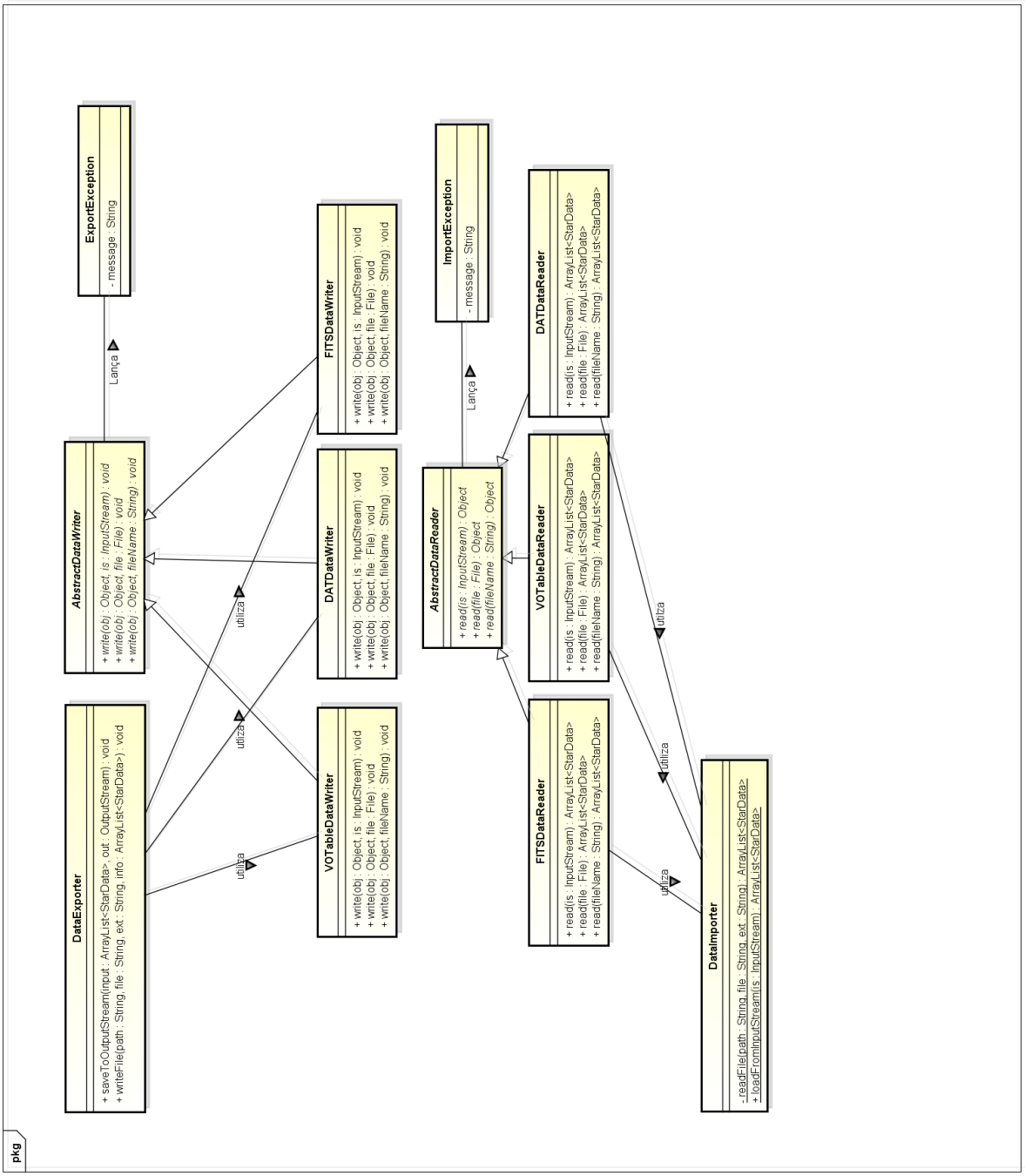


Figura 3-5 - Classes Presentes em Importar e Exportar Dados

AbstractDataReader é uma classe abstrata para importação de arquivos. *DATDataReader* é uma implementação de *AbstractDataReader* para fazer a leitura do arquivo de dados tipo texto DAT e transformá-lo em uma coleção de instâncias da classe *StarData*.

FITSDataReader e *VOTableDataReader* também são implementações de *AbstractDataReader* para importação dos formatos de arquivo FITS e VOTable respectivamente. Basicamente as classes que estendem *AbstractDataReader* precisam implementar os métodos *read(InputStreamis), read(File file)* e *read(StringfileName)*. Caso haja um erro na importação é levantada uma *ImportException* contendo o rastreio de exceções que levaram ao erro.

AbstractDataWriter é a contraparte de *AbstractDataReader* sendo uma classe abstrata para exportação de arquivos. Analogamente DATDataWriter, FITSDataWriter e VOTableDataWriter devem implementar os métodos *write(Objectobj, InputStreamis), write(Objectobj, File file) e write(Objectobj, StringfileName)*. Caso haja um erro na exportação é levantada uma exceção da classe ExportException contendo o rastreio de que levou ao erro.

Os pontos de entrada para Importação e Exportação são as classes *DataImporter* e *DataExporter* chamadas a partir do *ServletGraphServlet* não incluído neste diagrama. Ao realizar-se a importação de arquivos, as informações na forma de uma estrutura de *ArrayList* de objetos *StarData*, são então serializadas e colocadas em escopo de sessão para uso posterior.

3.4.3 Mapeamento do VOTable

No caso particular da importação de VOTable utilizou-se o a API xstream que converte XML para uma estrutura de classes e vice versa. Uma das exigências dessa API que as *tags* do XML sejam mapeadas como POJO´s, *Plain Old Java Objects*. A Figura 3-6 exemplifica esta estrutura. Este assunto é mais detalhado no Capítulo 4.

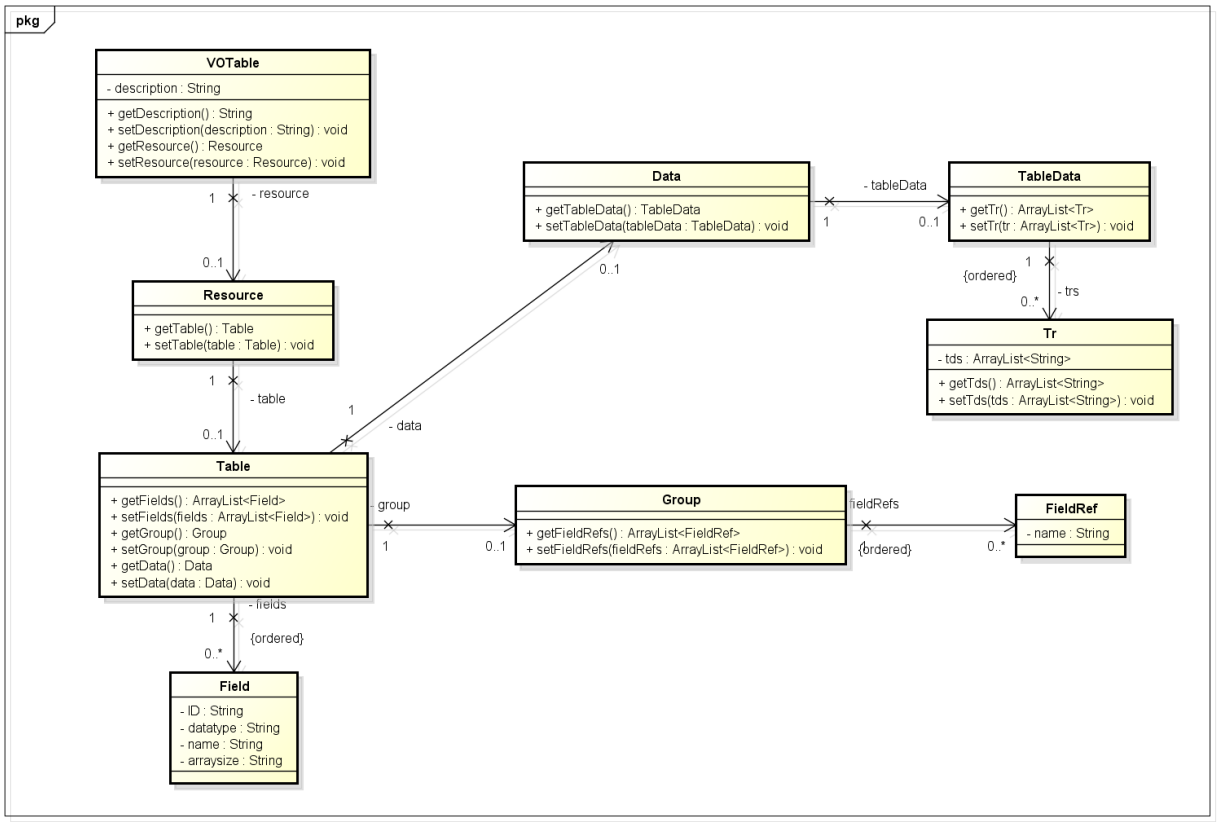


Figura 3-6 - Classes presentes no mapeamento de VOTable

4 Aspectos de Implementação

Neste Capítulo descrevemos aspectos técnicos referentes à implementação da ferramenta. A versão mais atual o sistema desenvolvido encontra-se hospedada e pode ser acessada por meio da seguinte URL: http://teammobile.kinghost.net/StellarCluster.

4.1 Arquivos de Dados

Diferentes tipos de arquivos de dados são utilizados para análises astronômicas. Cada um deles possui características próprias e diferentes finalidades. Nesta Seção, descrevemos os formatos de dados que nossa ferramenta manipula.

4.1.1 Arquivo texto DAT

4.1.1.1 Descrição do formato

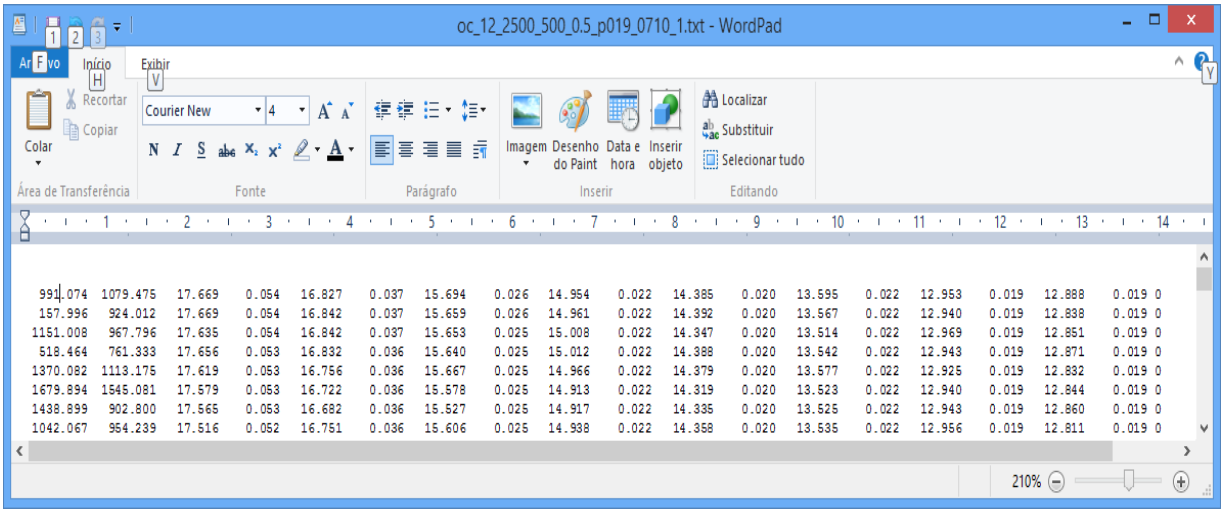
A ferramenta tem a capacidade de manipular dados sintéticos em formato de texto ASCII. Segundo informações fornecidas pelo astrônomo Alberto Krone-Martins, que produziu esses dados, cada arquivo de dados é formado por N linhas, cada uma referente a uma estrela do aglomerado ou uma estrela do campo, isto é, estrela da galáxia que está em frente ou atrás do aglomerado, mas, como não há informação de distâncias, não se pode diferenciá-las de forma simples. Estes arquivos foram elaborados de forma que a distribuição de probabilidades corresponde à simulação de um cenário específico. As colunas dos arquivos são:

x y U eU B eB V eV R eR I eI J eJ H eH K eKfield

As primeiras duas (x e y), são posições da estrela na imagem. As colunas U B V R I J H K são medidas da magnitude relativa, ou do brilho da estrela (ao contrário: quanto menor o valor, maior o brilho). A escala na qual este brilho está codificado é logarítmica.

As colunas eU eB eV eR eI eJ eH eK são indicadores do erro estimado durante a realização das medidas das colunas de magnitude equivalente (eU é o erro em U, e assim por diante.). A última coluna (field) é um indicador se a estrela pertence ao campo ou ao aglomerado. Assume o valor 0 indicando uma estrela do aglomerado e 1 indicando uma

estrela do campo. Obviamente, a existência deste campo nestes arquivos deve-se ao fato de tratarem-se de dados simulados, um exemplo desses arquivos é mostrado na Figura 4-1.



991.074	1079.475	17.669	0.054	16.827	0.037	15.694	0.026	14.954	0.022	14.385	0.020	13.595	0.022	12.953	0.019	12.888	0.019	0
157.996	924.012	17.669	0.054	16.842	0.037	15.659	0.026	14.961	0.022	14.392	0.020	13.567	0.022	12.940	0.019	12.838	0.019	0
1151.008	967.796	17.635	0.054	16.842	0.037	15.653	0.025	15.008	0.022	14.347	0.020	13.514	0.022	12.969	0.019	12.851	0.019	0
518.464	761.333	17.656	0.053	16.832	0.036	15.640	0.025	15.012	0.022	14.388	0.020	13.542	0.022	12.943	0.019	12.871	0.019	0
1370.082	1113.175	17.619	0.053	16.756	0.036	15.667	0.025	14.966	0.022	14.379	0.020	13.577	0.022	12.925	0.019	12.832	0.019	0
1679.894	1545.081	17.579	0.053	16.722	0.036	15.578	0.025	14.913	0.022	14.319	0.020	13.523	0.022	12.940	0.019	12.844	0.019	0
1438.899	902.800	17.565	0.053	16.682	0.036	15.527	0.025	14.917	0.022	14.335	0.020	13.525	0.022	12.943	0.019	12.860	0.019	0
1042.067	954.239	17.516	0.052	16.751	0.036	15.606	0.025	14.938	0.022	14.358	0.020	13.535	0.022	12.956	0.019	12.811	0.019	0

Figura 4-1 - Arquivo de Dados Sintéticos

4.1.1.2 Importação

Na importação do arquivo DAT por tratar-se de texto puro, exigiu uma técnica relativamente simples de importação. Utilizou-se a classe FileReader do JAVA especializada na leitura de arquivos ou *streams* de texto. Cada linha do arquivo é separada em cadeias de forma a se extrair a informação de cada coluna. Cada cadeia é então convertida no tipo de dado correspondente a coluna na ordem exata como mostra o trecho de código a seguir:

```
// separa strings pelo caracter espaco
String[] data = buffLine.split(" ");
star = new  StarData();
// verifica se primeiro elemento representa uma string nula
if (data[0] == null || data[0].equals("")) {
        index = 1;
}
star.setX(new Double(data[index])); index++;
star.setY(new Double(data[index])); index++;
star.setU(new Double(data[index])); index++;
```

4.1.2 Arquivos FITS

4.1.2.1 Descrição do formado

O formato *Flexible Image Transport System*(FITS) foi inicialmente desenvolvido por astrônomos nos EUA e na Europa no final de 1970 para intercâmbio de dados entre observatórios (THE FITS SUPPORT OFFICE, 2013). Ainda está sendo usado como um formato de intercâmbio de dados e arquivamento por astrônomos. Trata-se de um formato de arquivo projetado para armazenar, transmitir e manipular imagens e dados científicos associados. O termo *imagem* em nome do padrão é vagamente aplicada e arquivos FITS muitas vezes contêm apenas dados de não-imagem. Desde o seu início, FITS foi visto como um formato de transporte por mais de uma imagem estática destinada à impressão ou visualização em duas dimensões. FITS foi concebido para facilitar a transmissão isenta de erros de matrizes de dados de *n* dimensões. Estes *arrays* multidimensionais podem ser espectros 1- D, imagens 2-D ou cubos de dados de três ou mais dimensões. Tabelas de duas dimensões contendo linhas e colunas de dados também podem ser armazenadas em um arquivo FITS, por isso é classificado principalmente como um formato de conjunto de dados, com o uso de dados de imagem como secundário.

Quando o FITS é usado para dados de imagem simples, essa imagem possuem coordenadas tridimensionais (x ,y , cor) , com o terceiro eixo representando diferentes pontos de um eixo espectral, medida pelo comprimento de onda, frequência, ou qualquer outra escala adequada. Não há representação de espaço de cores como RGB.

FITS também é muitas vezes usado para armazenar dados não-imagem, como espectros , listas de fótons , cubos de dados , ou mesmo dados. Foi projetado para suportar uso de arquivamento de longo prazo.

4.1.2.2 Importação

A NASA, *National Aeronautics and Space Administration* mantém um site com as principais APIs para manipulação de arquivos FITS. No caso da linguagem JAVA uma das API´s listadas é a *nom.tam.fits*. Para simplificar a importação optou-se por usar o formato TableData apenas cuja utilização pode ser vista em um trecho de código da classe FITSHandler

```
Fits fitsFile = newFits(fits);

// obtem o HDU que contem a tabela
BasicHDU hdu1 = fitsFile.getHDU(1);

ColumnTable table = (ColumnTable)hdu1.getData().getData();
// loop nas linhas da tabela
```

```
for (int i = 0; i <table.getNRows(); i++) {
        StarData star = newStarData();

        // obtem a linha corrente
        Object[] row = (Object[]) table.getRow(i);


        star.setX(((double[]) row[0])[0]);
        star.setY(((double[]) row[1])[0]);
        star.setU(((float[]) row[2])[0]);
        star.seteU(((float[]) row[3])[0]);
....
}
```

4.1.3 Arquivo VOTable

4.1.3.1 Descrição do formato

O formato VOTable, *Virtual Observatory Table* (IVOA, 2013) é construído de código XML (*Extensible Markup Language*) e foi concebido como um armazenamento flexível e formato de troca de dados tabulares, com particular destaque para as tabelas astronômicas. A interoperabilidade é incentivada através do uso de padrões e o XML permite que os aplicativos possam validar facilmente um documento de entrada, bem como facilitar transformações através de XSLT (*eXtensible Style Language Transformation*).

VOTable foi construído com recursos para *Big Data* e *Grid Computing*. Ele permite que os metadados e dados possam ser armazenados separadamente. Os processos podem então usar estes metadados para se preparar para seus dados de entrada, ou para organizar de terceiros ou transferências paralelas dos dados. Assim como estamos acostumados a idéia de enviar um ponteiro para um documento (URL) no lugar do documento, pode-se agora enviar ponteiros de metadados de tabelas de dados no lugar das próprias tabelas. A parte de dados em uma VOTable pode ser representado usando um dos três formatos diferentes: TableData , FITS descrito anteriormente e BINARY. TableData apresenta-se num formato XML puro, de modo que pequenas tabelas podem ser facilmente manipuladas em sua totalidade por ferramentas que trabalhem com XML. Suporta também o formato de tabela binária do FITS para eficiência e facilidade de programação.

VOTable pode ser utilizado de maneiras diferentes, como um dispositivo de armazenamento de dados e de formatos de transporte, e também como um meio para armazenar os metadados sozinho, ou seja, apenas a estrutura da tabela. Neste último caso, pode-se imaginar uma estrutura VOTable sendo enviada para um servidor, o que pode , em seguida, abrir uma ligação de banda larga para receber os dados reais, utilizando a estrutura previamente digerido como uma forma de interpretar a corrente de bytes de dados soquete.

VOTable pode ser usado para pequenos números de pequenos registros (tabelas XML puras) , ou para um grande número de registros simples (*streaming* de dados) , ou ele pode ser usado para pequenos números de objetos maiores. Neste último caso , haverá software para espalhar grandes blocos de dados entre múltiplos processadores no Grid. Atualmente, a estrutura mais complexa que pode ser uma célula VOTable é uma matriz multidimensional.

Abaixo encontra-se a estrutura XML de um arquivo VOTable convertido de um dos arquivos DAT sintéticos. :

```
<?xml version='1.0'?>
<VOTABLE version="1.2"
xmlns="http://www.ivoa.net/xml/VOTable/v1.2">
<!--
 !  VOTableescrito a partir de oc_12_2500_500_0.5_p019_0710_1.dat
!-->
<RESOURCE>
<TABLE name="G:\Projeto Final\oc_12_2500_500_0.5_p019_0710_1.dat" nrows="5408">
<FIELD datatype="double" name="col1"/>
<FIELD datatype="double" name="col2"/>
<FIELD datatype="float" name="col3"/>
<!-- outras colunas de cabeçalho do dataset-->
<FIELD datatype="short" name="col19">
<VALUES null='-32768'/>
</FIELD>
<DATA>
<TABLEDATA>
<TR>
<TD>991.074</TD>
<TD>1079.475</TD>
<TD>17.669</TD>
<!-- outras colunas do dataset-->
<TD>0</TD>
</TR>
<TR>
<!-- demais linhas do dataset-->
</TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

4.1.3.2 Importação

O arquivo VOTable(IVOA, 2013)possui uma estrutura de XML, razão pela qual optou-se por utilizar a API *XStream* para conversão. *XStream* é uma biblioteca leve que permite ler um XML e obter uma estrutura de objetos. Da mesma forma é possível salvar uma estrutura de objetos em um arquivo XML. Para que isto aconteça é necessário mapear as *tags* e atributos como objetos e inicializar o *XStream* com este mapeamento. Se o arquivo XML é composto apenas de *tags* aninhadas que não possuem atributos, o mapeamento é simples. Este não foi o caso desta importação pois a especificação do VOTable contempla várias tags aninhadas cada uma contendo seus atributos.OMapeamento consiste na associação das classes JAVA já descritas no Capítulo 3 e na Figura 3-6. As respectivas *tags* no documento XML como mostra o trecho de código a seguir:

```
XStreamxstream = new XStream(new DomDriver());
xstream.alias("VOTABLE", VOTable.class);
xstream.alias("FIELD", Field.class);
xstream.alias("FIELDref", FieldRef.class);
xstream.alias("RESOURCE", Resource.class);
xstream.alias("TABLE", Table.class);
xstream.alias("TR", Tr.class);
xstream.alias("TD", String.class);
xstream.alias("TABLEDATA", TableData.class);
xstream.alias("DATA", Data.class);
xstream.alias("GROUP", Group.class);
xstream.aliasField("TABLE", Resource.class, "table");
xstream.aliasField("RESOURCE", VOTable.class, "resource");
xstream.aliasField("DESCRIPTION", VOTable.class, "description");
xstream.aliasField("TABLEDATA", Data.class, "tableData");

xstream.aliasField("DATA", Table.class, "data");
xstream.addImplicitCollection(Table.class, "fields");
xstream.addImplicitCollection(Group.class, "fieldRefs");
xstream.addImplicitCollection(Tr.class, "tds");
xstream.addImplicitCollection(TableData.class, "trs");
xstream.ignoreUnknownElements("VALUES");
```

Uma vez que o mapeamento é concluído, a importação do arquivo pode ser realizada por meio do comando:

```
Objectobj = xstream.fromXML(<nome do arquivo>);
```

O retorno é a instância de um objeto que corresponde a raiz do documento XML. Neste caso da classe VOTable. Observe que para que este mapeamento funcione é necessário criar uma classe para cada presente no documento além de mapear as coleções.

4.2 Construção da camada de apresentação

Optou-se por utilizar para construção da camada de apresentação da ferramenta o *Bootstrap* (Bootstrap, 2013). O *Bootstrap* é uma API de código livre inicialmente criada pelo *Twitter* e cujo objetivo é tornar as páginas responsivas. Isso significa que o layout de páginas WEB ajusta-se dinamicamente, tornando-o capaz de ser aberto em vários browsers e considerando as características do dispositivo usado (PC, tablet, celular) com um mínimo de codificação. O Bootstrap consiste em um conjunto de estilos CSS (*Cascade StyleSheets*) e arquivos Javascript com suporte ao HTML 5.

A implementação de algumas funcionalidades fornecidas pela ferramenta demandaram o uso do *framework Jquery*. Um exemplo disso é a funcionalidade que permite alterar a visualização e o comportamento do botão de upload como mostra a Figura 4-2.

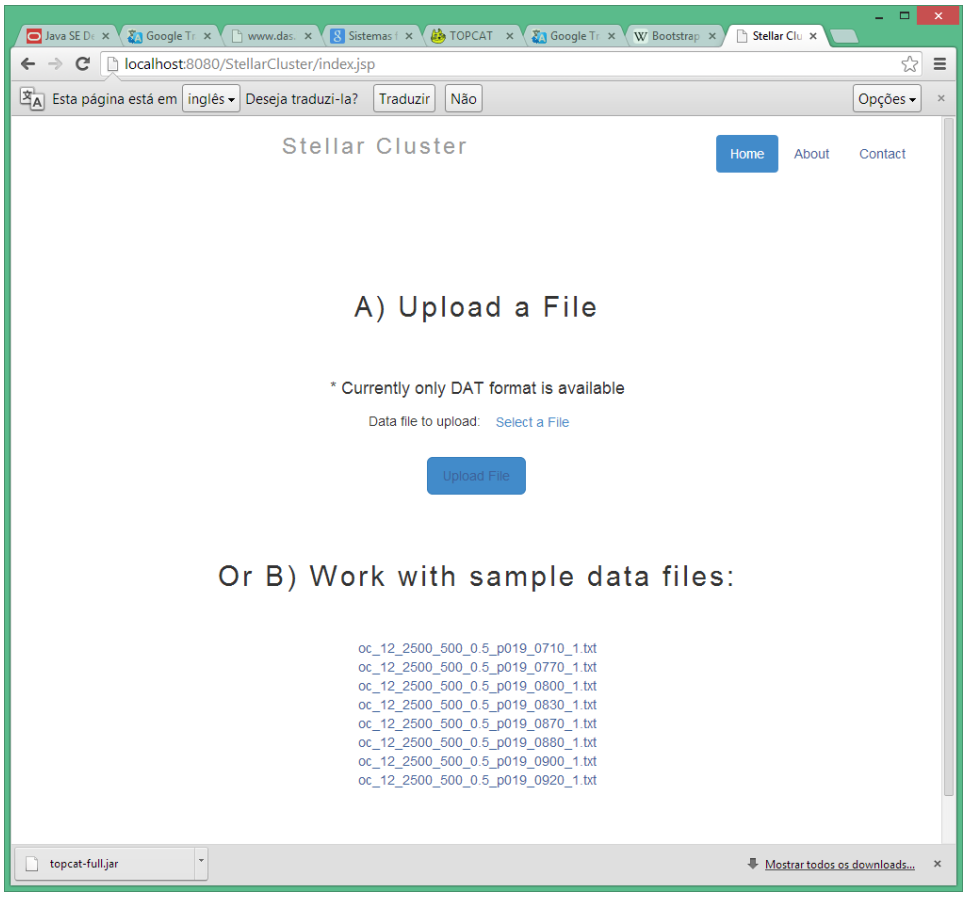


Figura 4-2 - Tela de entrada da aplicação criada com o Bootstrap.

4.3 Geração das Visualizações

O objetivo desta Seção é detalhar os aspectos relativos à geração das imagens e gráficos que são gerados pela ferramenta.

4.3.1 Geração de imagem 2D de estrelas

Uma vez que os dados de entrada possuem informações de coordenadas (x, y), além da magnitude visual V associada a cada ponto, utilizou-se esta informação para gerar imagens representativas de cada conjunto. Observe que como a magnitude esta representada em escala logarítmica, para representar cada objeto utiliza-se um ponto para valores iguais ou inferiores ao valor médio e cujo índice RGB é proporcional à magnitude visual V. Para valores superiores à média, desenham-se uma ou mais elipses cujos raios também são proporcionais a magnitude. Esta solução é análoga ao desenho de uma função gaussiana em 2D (UMM AL-QURA UNIVERSITY, 2014). Esta técnica apresenta um resultado próximo ao aspecto real do campo. Observa-se o resultado obtido na Figura 4-3.

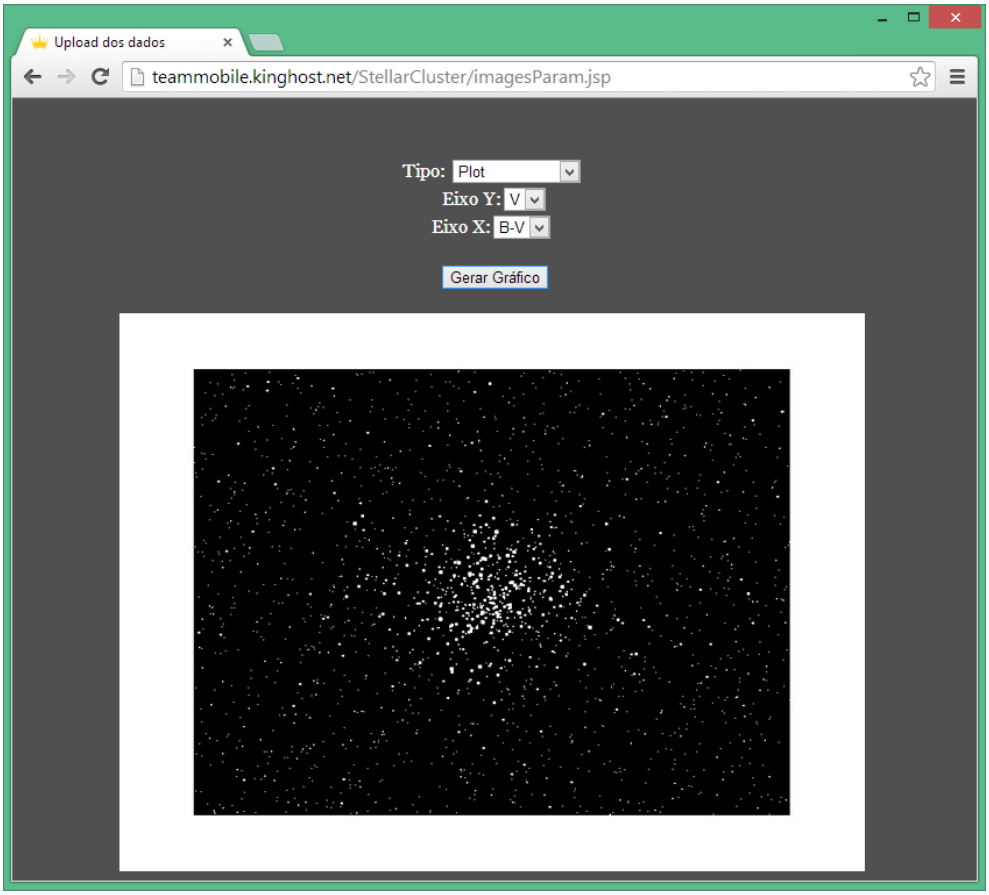


Figura 4-3 - Imagem 2D de estrelas gerada pelo sistema

4.3.2 Geração dos Diagramas Cor-Magnitude

O CMD é construído com a magnitude visual V como o eixo Y e um dos índices de cor como o eixo X. O *Processing* não possui uma funcionalidade para geração de gráficos a partir de um conjunto de dados. Por conta disso, os métodos tiveram que ser construídos utilizando-se os elementos simples disponibilizados pela API e no JAVA. Por padrão, somente a escala do eixo de magnitudes é desenhada. Pode ser atribuído um título ao gráfico localizado na parte superior. Um exemplo é mostrado na Figura 4-4.

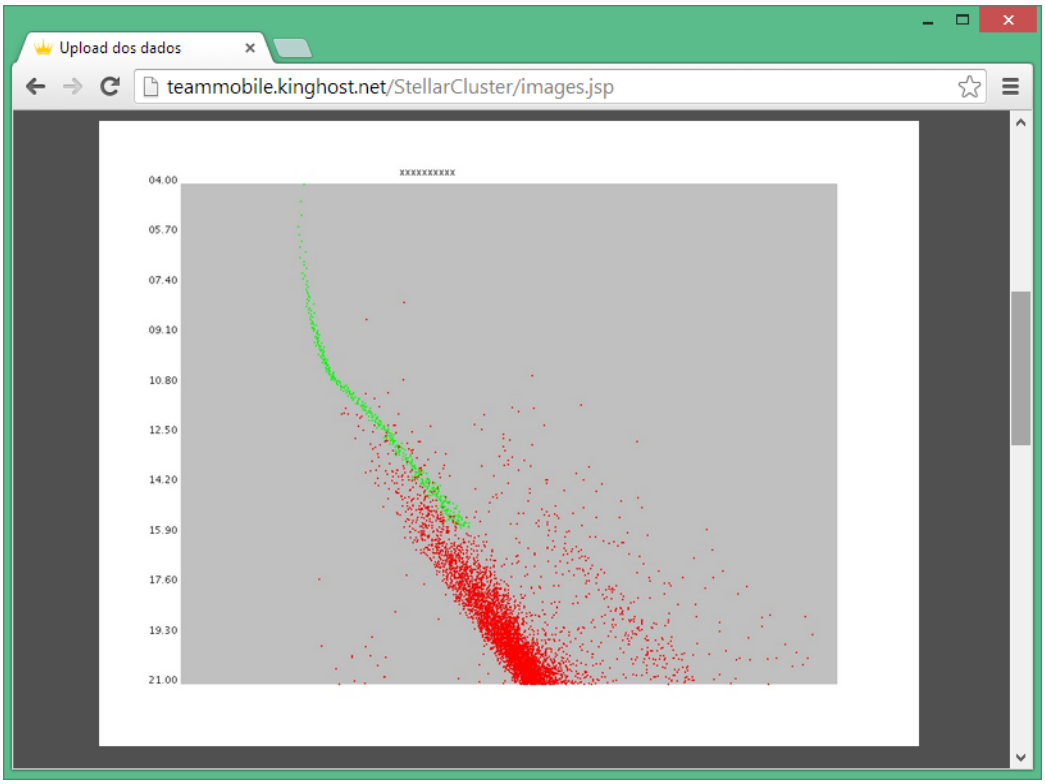


Figura 4-4 - Diagrama CMD gerado pelo sistema

4.3.3 Comunicação ente *Applet* e *Servlet*

A Comunicação entre a *Applet* e a aplicação é feita via *Servlet*. A *Applet* solicita ao *HTTPServlet* os dados em sessão. Foi criada uma classe *GraphCache* que segue o Padrão de projeto *Singleton*. Esta classe contém uma tabela *Hash* que representa o mapeamento entre o identificador da sessão do usuário e o dado carregado. O *Servlet* recupera por meio do atributo *sessionid* a sessão do usuário e com base nesse atributo serializa a coleção que contém a lista de objetos *StarData*. Para que isto aconteça é necessário que a classe *StarData* implemente a

interface *serializable*. Na *Applet* o conteúdo retornado é então desserializado obtendo-se assim a coleção de objetos da classe *StarData*.

4.3.4 Representação dos dados em três dimensões

Criar uma representação 3D no *Processing* requer passos a mais do que uma representação 2D. É necessário também configurar uma câmera para que o usuário possa interagir com o dado desenhado, a Figura 4-5 ilustra um cubo com os pontos de um conjunto de dados simulados. Sendo assim, as setas esquerda e direita movem o gráfico horizontalmente e as setas acima e abaixo o movem verticalmente. Em um computador IBM/PC Core 2 duo e placa de vídeo de 128 MB a visualização ocorreu sem apresentar quebra aparente de quadros.

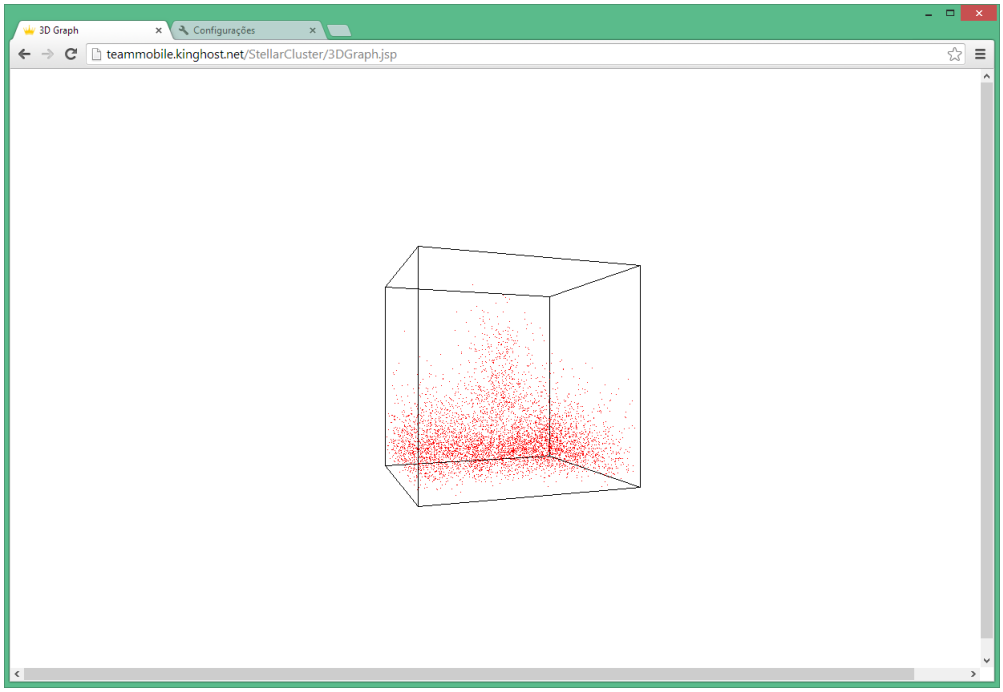


Figura 4-5 - Gráfico em três dimensões apresentando as coordenadas x, y e a magnitude visual V.

4.3.5 Funcionalidade Mosaico

Na funcionalidade de mosaico, os vários CMD´s gerados sobre um mesmo conjunto de dados estão dispostos lado a lado. Esta tela é interativa e opcionalmente pode-se selecionar um subconjunto dos dados por meio do mouse e esta seleção é destacada nos demais CMD´s e na representação Cartesiana XY. Esta seleção de pontos pode dar suporte a análise de comportamento das estrelas que pertencem ao aglomerado em diferentes espectros.

Por tratarem-se de dados simulados, optou-se por destacar na cor verde os objetos que pertencem ao aglomerado e em vermelho os que não pertencem a nenhum aglomerado como mostra a Figura 4-6.

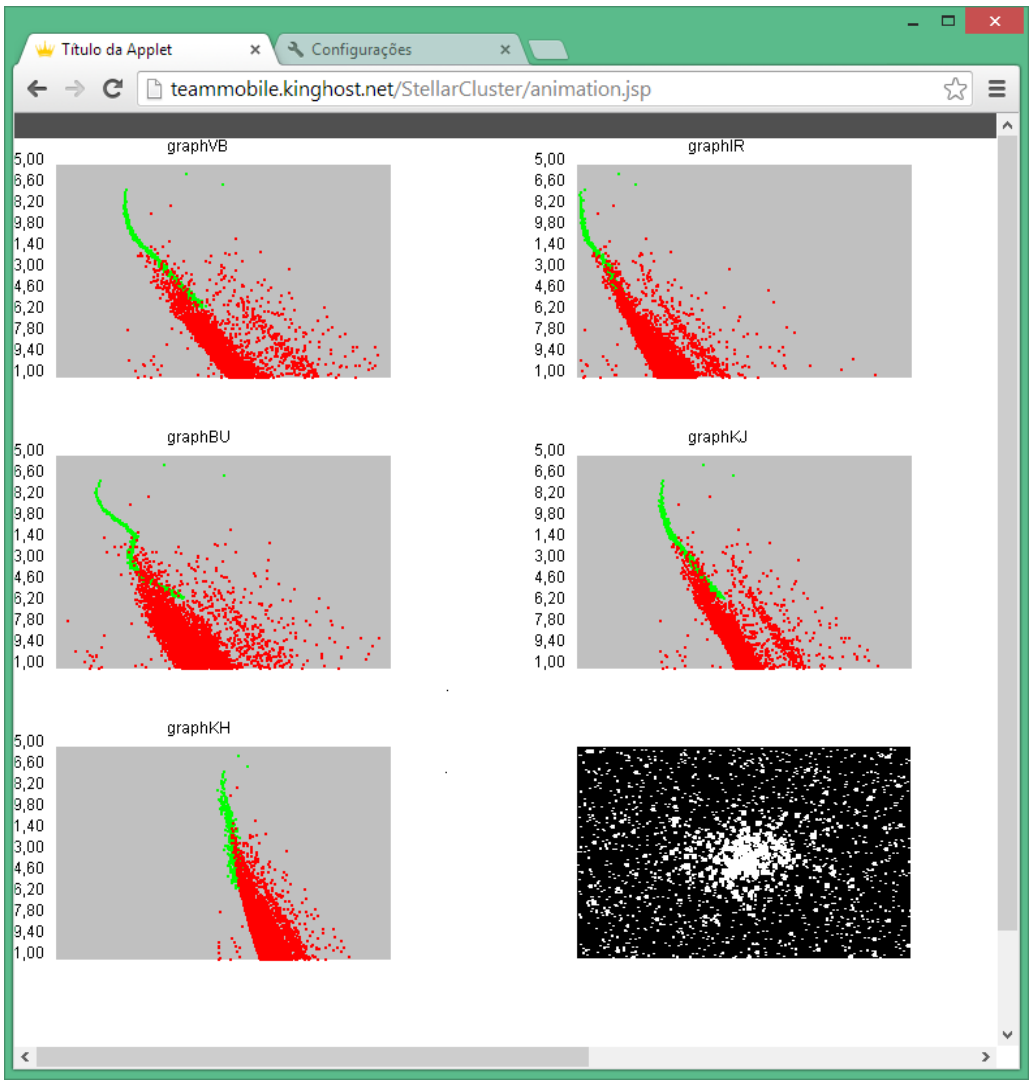


Figura 4-6 - Funcionalidade Mosaico

5 Conclusões

5.1 Considerações Finais

Foi desenvolvida como Projeto Final uma ferramenta WEB de código aberto para visualização e suporte a uma posterior análise exploratória de dados astronômicos.

Esta ferramenta lê arquivos do tipo FITS, DAT e VOTABLE, gera gráficos 3D, gráficos Cor-Magnitude e apresenta a imagem projetada 2D de conjuntos de estrelas. De forma geral, é uma ferramenta de visualização de fácil acesso que permite ao usuário explorar dados de estrelas e analisar que tipos de característica possuem as estrelas que fazem parte de um aglomerado.

Uma das dificuldades deste trabalho foi encontrar fontes de documentação adequadas correspondentes às teorias envolvidas e com relação à importação dos arquivos de formatos FITS e VOTable. A estratégia adotada foi simplificar o problema, selecionando, dentro da abrangência de opções, as que mais se adequavam ao escopo desta ferramenta.

Outra dificuldade encontrada foi estabelecer a comunicação entre *Applet* e *Servlet* de forma satisfatória, pois a documentação oficial não é muito abrangente e existem nuances quanto a máquina virtual JAVA e as versões dos navegadores suportados.

5.2 Trabalhos Futuros

Dentre as melhorias propostas para este sistema está a de importar dados diretamente de um conjunto de imagens onde cada uma delas foi obtida por meio de um filtro UBV.

Do ponto de vista da construção da aplicação pode ser feito um refatoramento para incluir um *framework* de injeção de dependências com o *Spring* e um *framework* para monitoramento e *log* como o *log4j*.

Outra melhoria proposta seria a de trabalhar com subconjuntos de dados como o software TOPCAT que permite aglutinar colunas e criar outras com base em alguns critérios ou fórmulas. Inicialmente pensou-se em usar um mecanismo de linguagem de expressão.

A identificação automática de aglomerados ficou fora do escopo deste projeto, mas espera-se que com os tipos de visualizações criados neste presente sistema, seja possível facilitar a análise visual de características que diferenciem estrelas de aglomerados. Além disso, a criação de novas funcionalidades e a participação de usuários com interesse na resolução deste problema pode manter o desenvolvimento deste projeto.

Referências

AMATEUR ASTRONOMERS, INC. The HR DIAGRAM, 2013. Disponivel em: <http://www.asterism.org/tutorials/tut39%20HR%20Diagram.pdf>. Acesso em: 29 jul. 2013.

ASTRONOMY ONLINE. Color Magnitude Diagram, 2013. Disponivel em: <http://astronomyonline.org/Astrophotography/CMDDiagram.asp>. Acesso em: 01 dez. 2013.

BOOTSTRAP , 2013. Disponivel em: <http://getbootstrap.com/>. Acesso em: 29 dez. 2013.

CLAUDIA.RODRIGUES. Fotometria, 2011. Disponivel em: <http://www.das.inpe.br/~claudia.rodrigues/ast203/fotometria.pdf>. Acesso em: 15 jan. 2014.

COSMOS, C. Distance Measure, 2014. Disponivel em: <http://coolcosmos.ipac.caltech.edu/cosmic_classroom/cosmic_reference/distance.html>. Acesso em: 15 fev. 2014.

DEPARTAMENTO DE ASTRONOMIA DA UFRGS. Aglomerados Estelares, 2014. Disponivel em: <http://www.if.ufrgs.br/oei/hipexpo/aglomerados.pdf>. Acesso em: 15 jan. 2014.

EAGLE, B. Star Classification - A Level Physics, 2013. Disponivel em: <https://www.youtube.com/watch?v=1tGOisbSSrw>. Acesso em: 11 ago. 2013.

FORBES, G. **History of Astronomy**. [S.l.]: University of Cambridge, 2009.

GALAXY Photography, 2014. Disponivel em: <http://www.galaxyphoto.com/>. Acesso em: 15 fev. 2014.

IMAGE REDUCTION AND ANALYSIS FACILITY. IRAF Home Page, 2013. Disponivel em: <http://iraf.noao.edu/>. Acesso em: 29 jul. 2013.

IVOA. VOTable Format Definition, 2013. Disponivel em: <http://www.ivoa.net/documents/VOTable/>. Acesso em: 26 dez. 2013.

MORGAN, W. W. . K. P. C. A. K. E. **An atlas of stellar spectra, with an outline of spectral classification**. Chicago: The University of Chicago Press, 1943.

RIDPATH, I. **Guia Ilustrado Zahar de Astronomia**. 1ª Edição. ed. [S.l.]: Zahar, 2007.

SAN DIEGO STATE UNIVERSITY. Wien's Law, 2013. Disponivel em: <http://mintaka.sdsu.edu/faculty/wfw/CLASSES/ASTROBIO/LECTURES/wien_law.pdf>. Acesso em: 28 dez. 2013.

TAYLOR, M. TOPCAT, 2013. Disponivel em: <http://www.star.bris.ac.uk/~mbt/topcat/>. Acesso em: 28 dez. 2013.

THE FITS SUPPORT OFFICE. The FITS Support Office, 2013. Disponivel em: <http://fits.gsfc.nasa.gov/>. Acesso em: 26 dez. 2013.

TODD, D. P. **Astronomy:** The Science of Heavenly Bodies. 2. ed. [S.l.]: Harpers and Brothers, 1922.

UMM AL-QURA UNIVERSITY. Gaussian Funcion, 2014. Disponivel em: <https://uqu.edu.sa/files2/tiny_mce/plugins/filemanager/files/4282164/Gaussian%20function.pdf>. Acesso em: 16 jan. 2014.

ZOMBECK, M. V. **Handbook of Space Astronomy and Astrophysics**. 2ª. ed. [S.l.]: Cambridge University Press, 1990.

Apêndice A - Manual do Usuário

A versão mais atual do software encontra-se atualmente hospedada em http://teammobile.kinghost.net/StellarCluster. Acesse esta URL preferencialmente usando um browser que suporte HTML5 como o Mozilla Firefox 20+ ou Google Chrome.

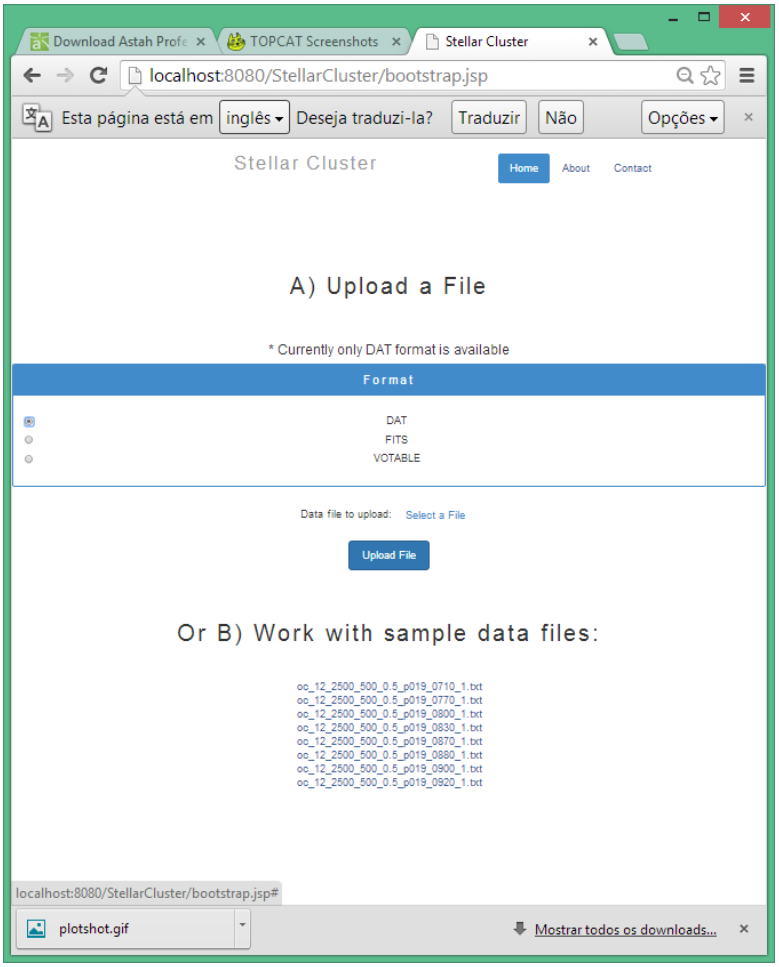


Figura A- 1 - Tela principal do sistema.

Na tela principal (Figura A- 1) serão abertas duas opções:

A) "Upload File" onde o usuário tem a opção de enviar um arquivo de dados. Os formatos suportados são DAT (ASCII), VOtable e FITS. Basta clicar no link '*Select a File*' para abrir a caixa de diálogo que permite selecionar um arquivo localizado no computador do usuário. Seleciona-se o tipo correspondente ao arquivo que será enviado e clica-se no botão "*Send File*".

B) "*Work with sample data files*" onde o usuário trabalha com um dos conjuntos de dados pré-definidos. Basta clicar em um dos links apresentados na página e o conjunto será carregado.

Com os dados carregados, seja por envio de arquivo ou pelo uso da opção de conjunto de dados pré-definido, será apresentada uma tela (Figura A- 2) com as seguintes funcionalidades:

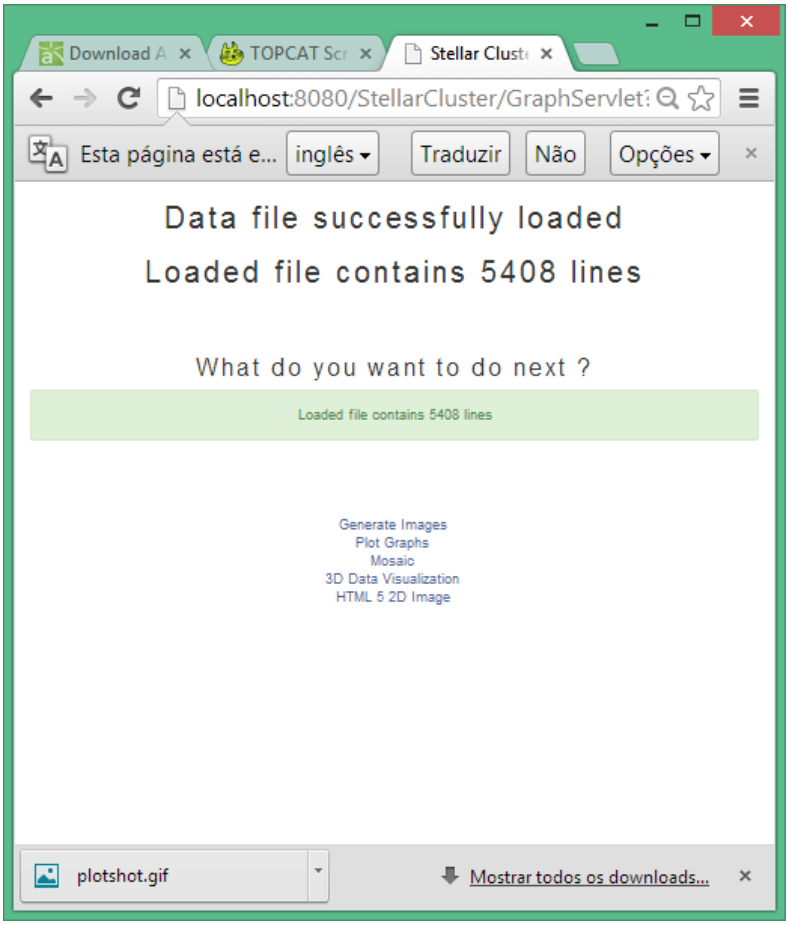


Figura A- 2 - Tela com as funcionalidades de visualização.

- Opção *Generate images:* Ao clicar nesta opção, são geradas imagens com opções pré definidas de estrelas 2D, e dois CMDs com magnitude V versus índice de cor U-B e B-V respectivamente (Figura A- 3).

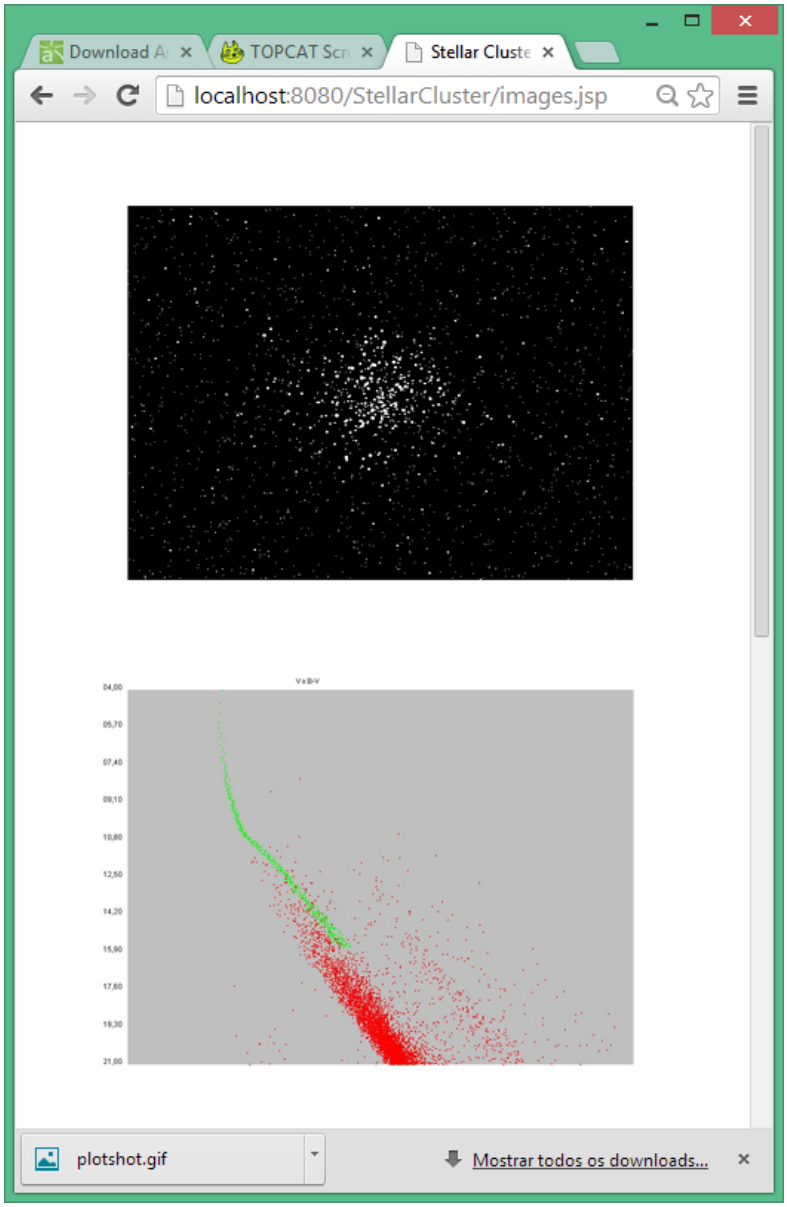


Figura A- 3 - Tela de projeção 2D das estrelas com dois diagramas CMD.

- Opção *Plot images:*Ao clicar nesta opção, o usuário é redirecionado para uma tela onde é exibido um formulário de opções para geração de imagens. Dentre as opções disponíveis estão estrelas 2D, HR-CMD, e HR-CMD colorizado. Para cada opção pode existir mais opções de geração (Figura A- 4).

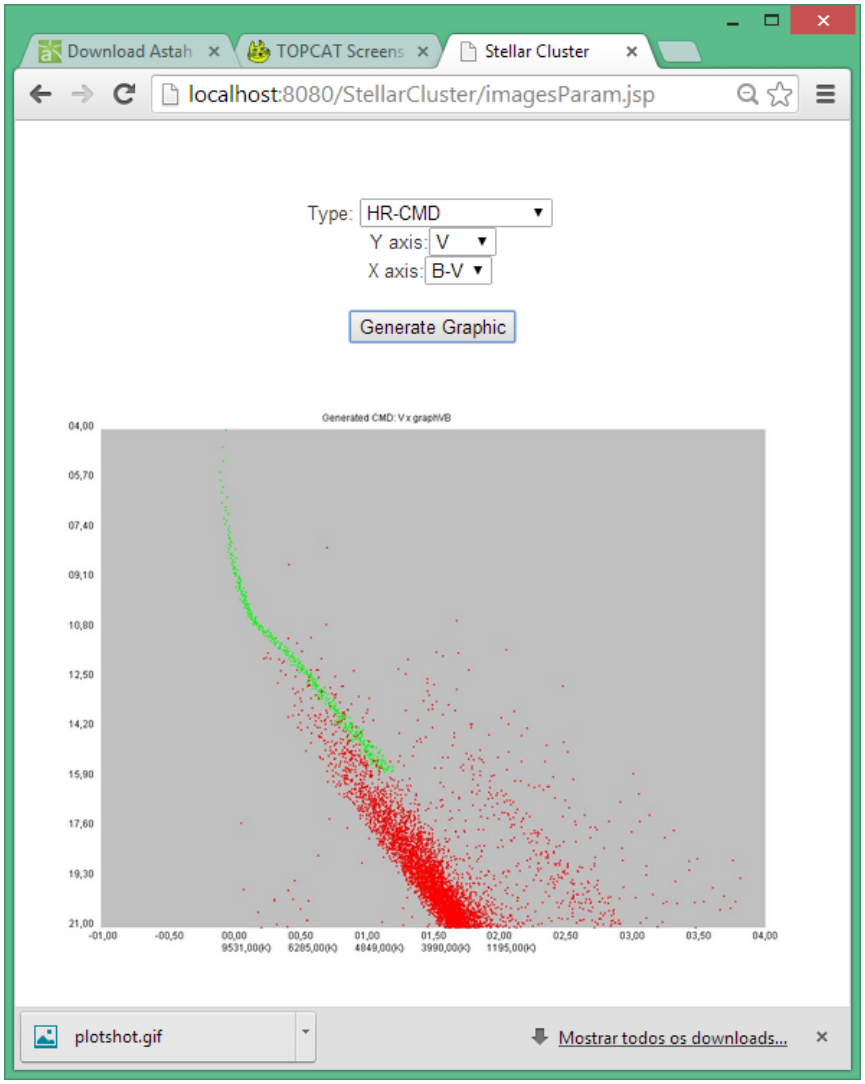


Figura A- 4 - Tela de geração de imagens personalizadas de acordo com os parâmetros selecionados.

- Opção *Mosaic:* Ao clicar nesta opção, são gerados 5 gráficos CMD com opções pré-definidas e mais um desenho de estrelas 2D. Esta tela está ilustrada na Figura A- 5 e permite selecionar um subconjunto de pontos utilizando o mouse em um dos CMD´s e estes mesmos pontos são destacados nos demais diagramas do mosaico. Isso facilita a análise do comportamento das estrelas de um aglomerado nos vários espectros.

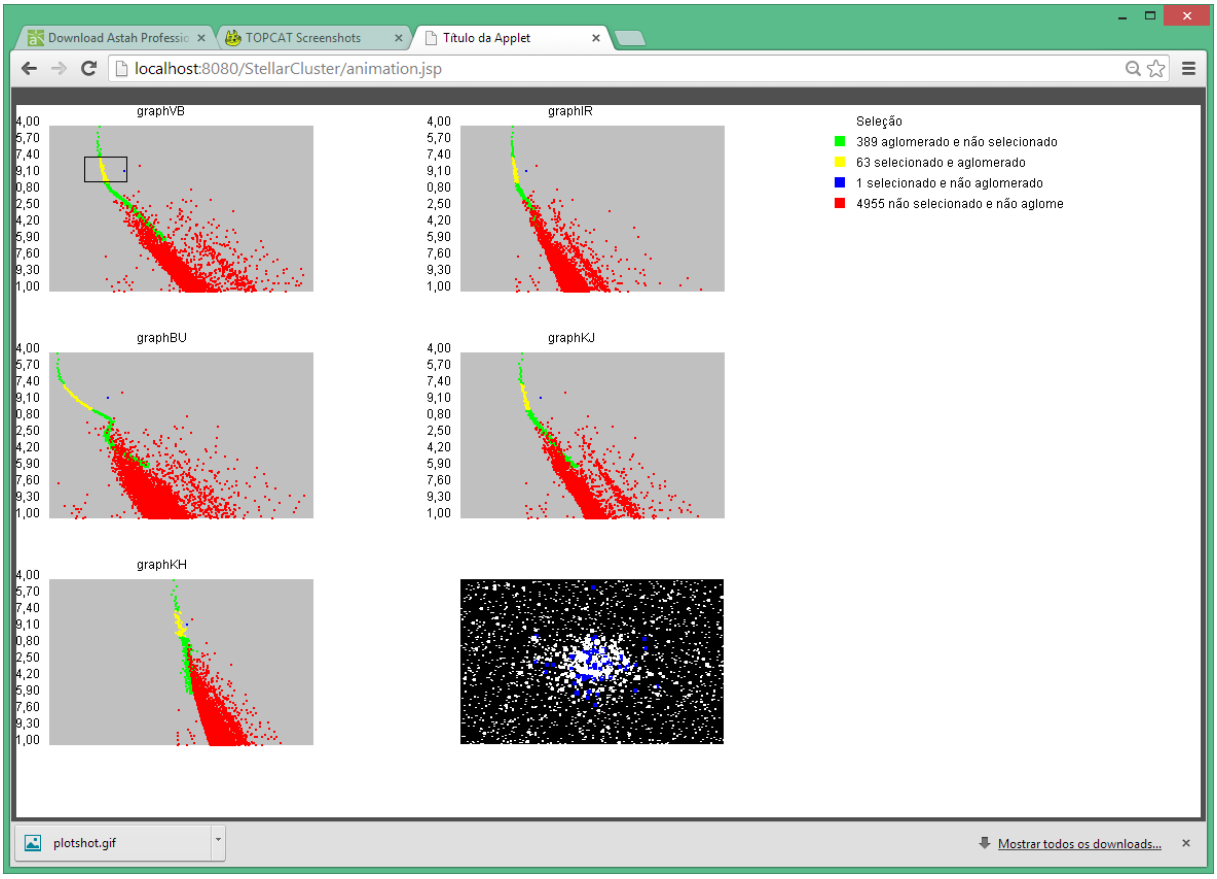


Figura A- 5 - Tela que mostra a funcionalidade mosaico e a seleção de um subconjunto sendo refletida nos demais diagramas com a cor amarela e na imagem 2D em azul.

-Opção *3D Data Visualization*: Ao clicar nesta opção, é criada uma visualização 3D dos dados arbitrariamente com os eixos X e Y como correspondentes as informações da colunas X e Y do conjunto de dados e o eixo Z como a magnitude aparente ou visual V (Figura A- 6).

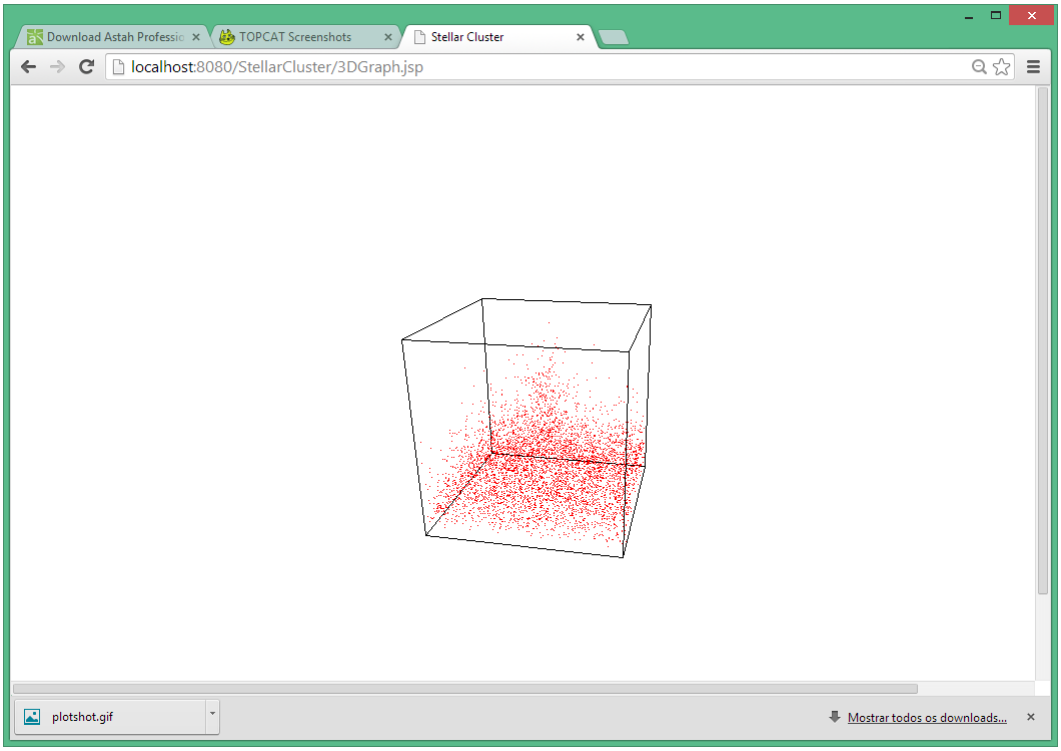


Figura A- 6 - Tela que mostra um cubo com pontos que representam as posições X e Y associadas a magnitude aparente.

-Opção *HTML 5 2D Image:* Ao clicar nesta opção, são geradas estrelas 2D. Para abrir esta opção não é necessário *Applets*, mas sim suporte a HTML 5, veja na Figura A- 7.

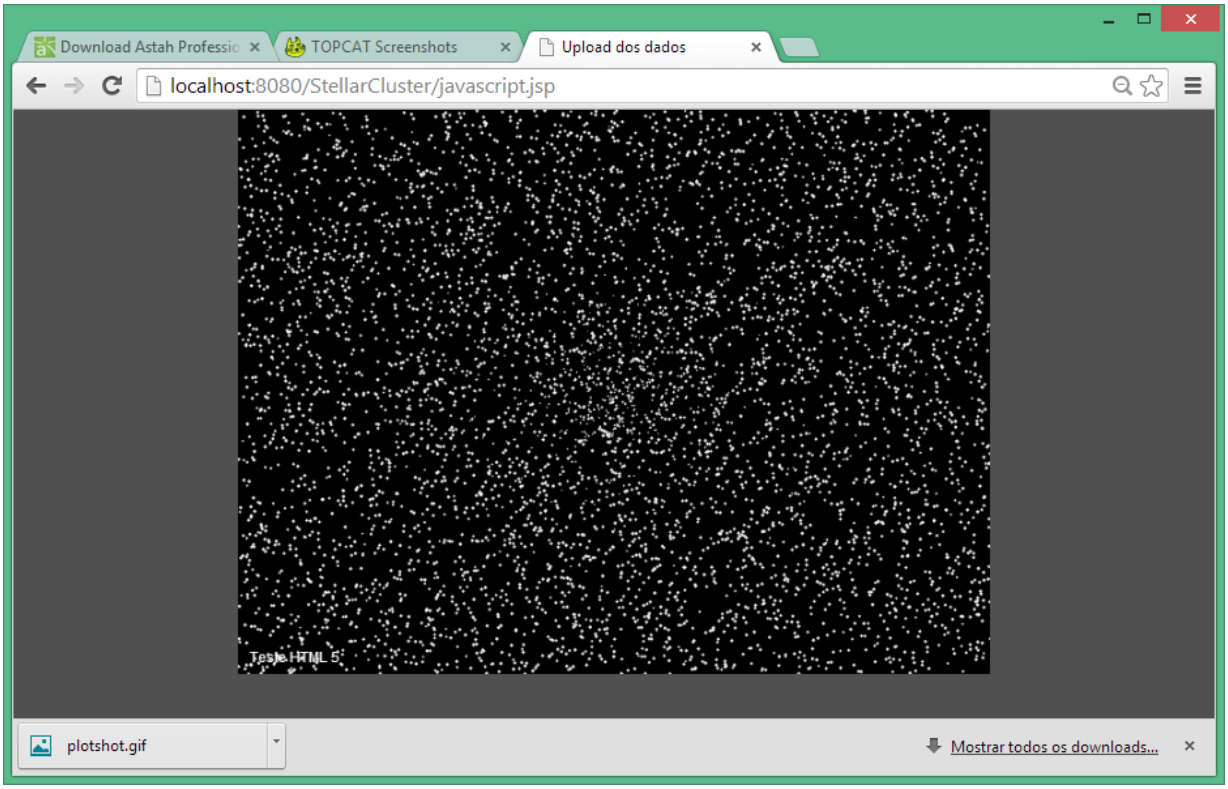


Figura A- 7 - Imagem 2D que representa o brilho de cada estrela de um conjunto de dados, utilizando recursos do HTML5.