



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO  
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**DESENVOLVIMENTO DE ROTINAS EM AMBIENTE MATLAB  
PARA O PROCESSAMENTO DE DADOS ÓPTICOS COLETADOS  
EM AMBIENTES AQUÁTICOS PARA APLICAÇÕES EM  
SENSORIAMENTO REMOTO DE ÁGUAS DE INTERIOR**

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA  
(PIBIC/CNPQ/INPE)**

Jhonisson Gabriel Cunha Souza de Sá (Fatec São José dos  
Campos, Bolsista PIBIC/CNPq)

E-mail: [jhonisson.gabriel@gmail.com](mailto:jhonisson.gabriel@gmail.com)

Claudio Clemente Faria Barbosa (DPI/OBT/INPE, Orientador)

E-mail: [claudio.barbosa@inpe.br](mailto:claudio.barbosa@inpe.br)

Julho de 2017



## RESUMO

Este trabalho tem como objetivo integrar em uma única plataforma os sistemas de correção de dados de três equipamentos: ACS, TRIOS e Hydroscat, aperfeiçoar os sistemas afim de deixar o código com melhor performance tanto no processamento quanto de consumo de memória e tornar os sistemas mais amigável para os usuários. As principais ferramentas utilizadas foram um computador, o software Matlab no qual todos os sistemas estão baseados e os dados brutos extraídos dos três equipamentos de análises mencionados anteriormente. Os métodos de gerenciamento de tarefas como o MVC (model-view-controller), que permite a divisão das tarefas, e o Scrum, que gerencia o tempo e execução de tarefas durante o projeto, foram tão importantes quanto os métodos especializados para o sistema como o teste caixa cinza, que auxiliou no entendimento e procedimento a serem executados em algumas tarefas, generalização de scripts, interação humano computador, teste unitário, padronização de documentação ou mesmo análise de tempo de resposta. O sistema ficou mais robusto, porém lento ao ser inicializado e utilizado pela primeira vez durante o dia, já que o software não está carregado ainda na memória, entretanto seus resultados durante o decorrer de um período trabalho se mostraram muito próximos aos valores dos sistemas antes a integração, levando em consideração que este novo sistema possui além das funções padrões já existentes possui também técnicas de interação humano computador o que torna a utilização mais amigável. O sistema é confiável, fácil de utilizar e fazer manutenção, possui estrutura padronizada e independente. Todas estas características integradas tornam a rotina do usuário mais produtiva e menos trabalhosa nas tarefas a que se referem ao projeto.

Palavras-chave: Usabilidade. Desenvolvimento. Padronização. Rotinas. Integridade.

## SUMÁRIO

	<b><u>Pág.</u></b>
1	INTRODUÇÃO..... 1
2	OBJETIVOS DO TRABALHO ..... 3
2.1.	Objetivo Principal..... 3
2.2.	Objetivos Específicos..... 3
3	MATERIAIS E MÉTODOS UTILIZADOS ..... 5
3.1.	Aperfeiçoamento do Algoritmo..... 5
3.2.	Aperfeiçoamento das Rotinas ..... 5
3.3.	Plataforma para as Rotinas ..... 6
3.4.	Usabilidade ..... 7
3.5.	Metodologia de Desenvolvimento ..... 8
3.5.1.	MVC ..... 8
3.5.2.	Scrum..... 8
3.5.3.	Teste Unitário e Teste de Integração ..... 9
3.5.4.	Teste de Caixa Branca e Teste de Caixa Cinza..... 9
4	ANÁLISES E RESULTADOS..... 11
4.1.	Familiarização ACS, Hydroscat e Trios ..... 11
4.2.	Identificação dos Pontos a serem Desenvolvidos nas Rotinas..... 11
4.3.	Desenvolvimento das Modificações..... 12
4.4.	Geração de Documentação do Código Desenvolvido ..... 13
5	CONCLUSÃO ..... 15
	REFERÊNCIAS BIBLIOGRÁFICAS ..... 17

## LISTA DE FIGURAS

	<b><u>Pág.</u></b>
Figura 3.1 - Fluxograma de Processo de uma rotina .....	6
Figura 3.2 – Tela inicial da plataforma .....	7
Figura 3.3 – Funções opcionais do usuário.....	8
Figura 3.3 – Realização de Teste.....	10
Figura 4.1 – Script com informações gerais .....	12
Figura 4.2 – Interface gráfica com dicas e informações .....	13

## **LISTA DE SIGLAS E ABREVIATURAS**

INPE	Instituto Nacional de Pesquisas Espaciais
LabISA	Laboratório de Instrumentação de Sistemas Aquáticos
ACS	Absorption and Attenuation meters
MVC	Model-View-Controller



## **1 INTRODUÇÃO**

O grande volume de dados que são retornados de uma análise por um equipamento de precisão normalmente usados nas pesquisas de campo é imenso para que um ser humano possa analisar sem nenhum tipo de equipamento de computação, é neste momento que um conjunto de funções e algoritmos é criado para mostrar de forma analítica ao ser humano uma resposta. Com o objetivo de aperfeiçoar o processamento de dados gerado pelos equipamentos ACS (Wetlabs 2016), TRIOS (Trios 2016) e HYDROSCAT (Hydroscat 2016), foram analisados códigos, rotinas e processos para melhorar o desempenho durante o processamento dos dados e facilitar a usabilidade das rotinas através do usuário.





## **2 OBJETIVOS DO TRABALHO**

### **2.1. Objetivo Principal**

Existe a possibilidade de aperfeiçoar as rotinas já desenvolvidas para produzir uma melhor performance quanto a sua utilização pelos usuários quanto pelo uso dos recursos físicos (hardware).

### **2.2. Objetivos Específicos**

- a) Aperfeiçoar ferramental já desenvolvido previamente em linguagem Matlab para o processamento de dados obtidos pelos equipamentos;
- b) Desenvolver interface gráfica Matlab para as rotinas desenvolvidas;
- c) Desenvolver em linguagem Matlab rotinas para processamento de dados ópticos dos equipamentos de medidas de distribuição de tamanhos de partículas em suspensão, além da geração de arquivos do simulador Hydrolight coletados em campanhas de campo para os dados dos equipamentos.



### **3 MATERIAIS E MÉTODOS UTILIZADOS**

#### **3.1. Aperfeiçoamento do Algoritmo**

As variáveis possuem o espaço já determinado antes de ser utilizada em um *looping* para que não seja necessário fazer a realocação de memória durante cada loop que é executado (isso reduz o tempo de processamento que seria gasto fazendo realocação).

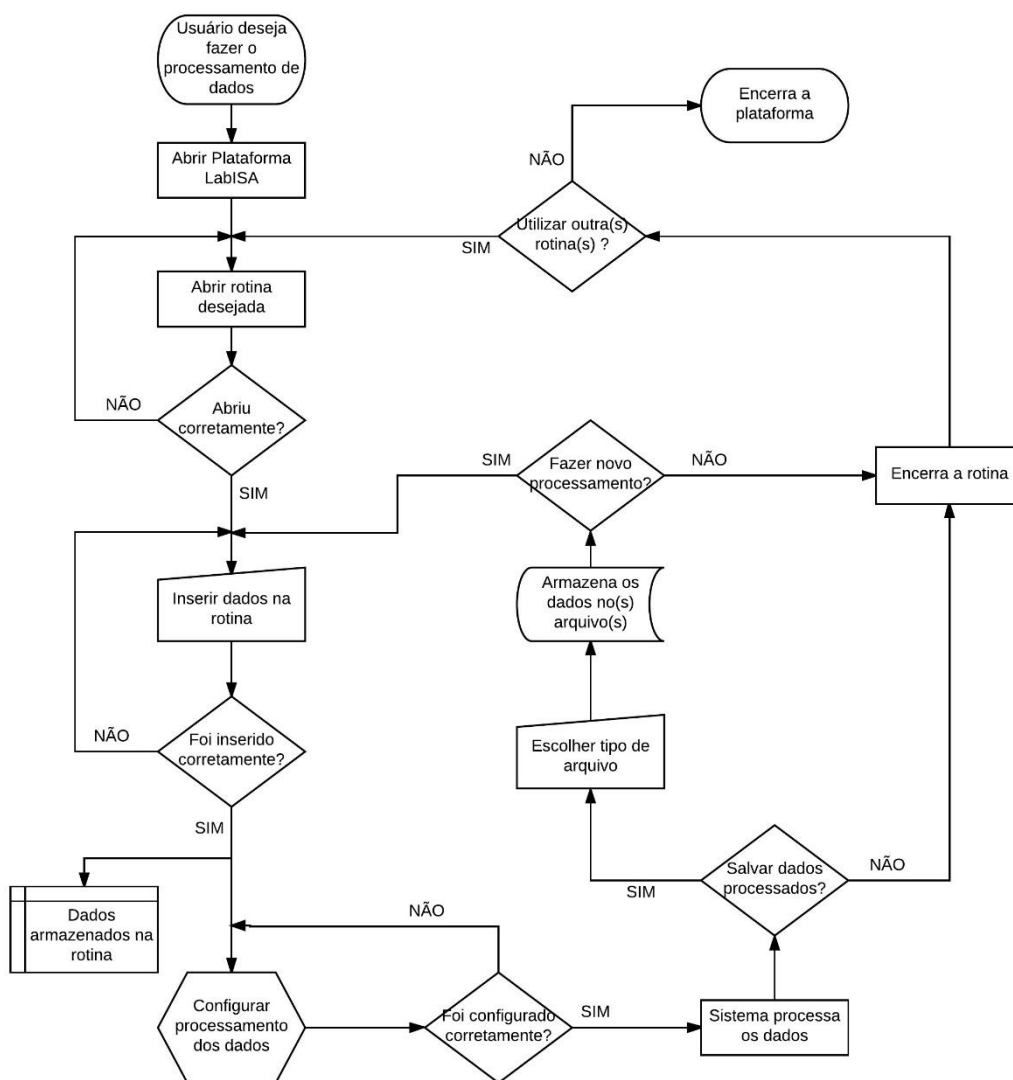
#### **3.2. Aperfeiçoamento das Rotinas**

As funções, scripts e processos que antes executavam várias funções independentes foram desmembrados e separados de pacotes com especificações semelhantes, auxiliando no momento de manutenção da rotina ou implementação de nova função.

Em cada função, script existem informações relevantes à rotina de modo geral. Informa a data da última atualização, se está ou não funcionando, informações sobre sua principal função na rotina, se é um script ou função, o significado da abreviação do nome, em qual pacote está localizado, em qual a posição ou ordem será executada na rotina, se possui objetos da interface gráfica relacionados e quais são. Esta outra boa prática de programação permite que o analista não necessite ler todo o código para entender o que faz e o que está contido nele.

Durante o progresso das rotinas os dados são armazenados de forma individual, toda e qualquer alteração feita em etapas anteriores que não forem confirmadas serão descartadas (analogamente pode ser comparado ao ROLLBACK de um sistema de gerenciamento de banco de dados que volta ao estado anterior a alteração por não ter finalizado com o COMMIT) mantendo a integridade dos dados.

Figura 3.1 - Fluxograma de Processo de uma rotina



Os fluxogramas de processo das rotinas de um modo geral são muito similares, a cada processo existe uma decisão para confirmar se a tarefa está seguindo de forma correta.

Fonte: Produção do autor.

### 3.3. Plataforma para as Rotinas

A plataforma serve como inicialização e integração entre as rotinas facilitando no momento de utilizá-las (não havendo ao usuário a necessidade de saber em qual diretório está localizado o arquivo que inicia uma determinada rotina).

Figura 3.2 – Tela inicial da plataforma



Fonte: Produção do autor.

A plataforma não se relaciona diretamente com um computador específico (computador em que foi desenvolvido) ou seja pode ser utilizado na maioria dos hardwares populares sem fazer alterações.

### **3.4. Usabilidade**

Os sistemas utilizam meios visuais para informar ao usuário, de forma intuitiva, se suas escolhas estão corretas ou não, bloqueios e desbloqueios de botões mostram quando pode prosseguir. Mensagens geradas em caixa de texto mostram informações mais relevantes e que requer certa atenção.

O usuário tem a opção de visualizar algumas informações ou não (antes o usuário era obrigado a visualizar).

Figura 3.3 – Funções opcionais do usuário



A opção Visualizar Gráfico é uma função que fica a critério do usuário utilizar ou não. Isso permite ao usuário acelerar o processo ou não visualizar informação que não são úteis para ele.

Fonte: Produção do autor.

### 3.5. Metodologia de Desenvolvimento

#### 3.5.1. MVC

O desenvolvimento da plataforma foi baseado no padrão de arquitetura MVC (model-view-controller), seu padrão criou facilidade na manutenção dos scripts por estar organizado em controladores, modelos e visualizadores e de forma independente umas das outras.

#### 3.5.2. Scrum

Durante o projeto procurou-se utilizar ao máximo a metodologia Scrum, por ser ágil e simples, a maioria das etapas eram padronizadas: entender

o código, identificar melhorias a serem desenvolvidas e desenvolver as melhorias, e caso surgissem obstáculos durante o desenvolvimento solucioná-los. Esta metodologia permitiu obter uma previsão aproximada de término do projeto e manter o foco em uma determinada tarefa, impedindo a dispersão.

### **3.5.3. Teste Unitário e Teste de Integração**

A cada tarefa desenvolvida utilizava o teste unitário para confirmar a funcionalidade individual e quando terminado todas as tarefas utilizava o teste de integração para confirmar funcionamento de todas em conjunto. Muitas precisaram ser alteradas pois algumas melhorias alteraram os parâmetros de entrada e saída.

### **3.5.4. Teste de Caixa Branca e Teste de Caixa Cinza**

O teste de caixa branca foi utilizado no início do projeto para identificar todas as etapas do processo das rotinas que já haviam sido desenvolvidas com o propósito de aperfeiçoar os algoritmos e lógicas dos scripts e funções, os novos scripts e funções após desenvolvidos utilizaram o teste de caixa cinza para verificar os resultados obtidos com valores de dados reais (supondo que o arquivo não tenha sofrido nenhuma alteração após ser gerado pelo equipamento).



Figura 3.3 – Realização de Teste



Esta figura mostra o momento do teste em que um parâmetro incorreto é inserido (número de cor vermelha). O botão que permite continuar o processo (botão *Próximo*) é desabilitado automaticamente até que o erro seja corrigido.

Fonte: Produção do autor.

## **4 ANÁLISES E RESULTADOS**

### **4.1. Familiarização ACS, Hydroscat e Trios**

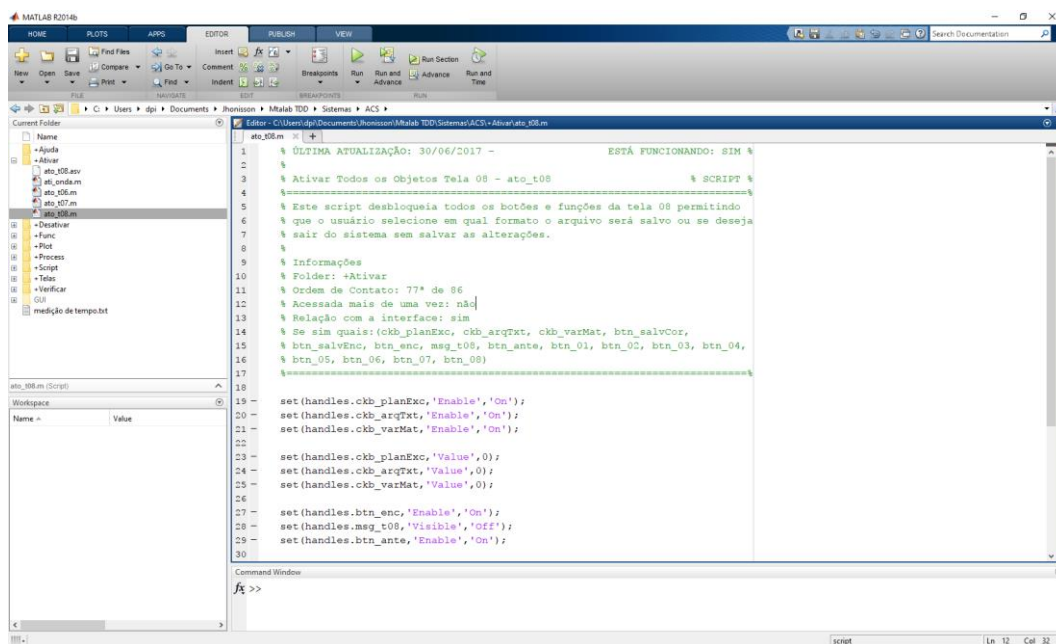
Um dos pontos mais importantes durante o desenvolvimento de uma ferramenta (software) é compreender cada parte do código, isto é essencial para saber a funcionalidade de cada linha de código, identificar erros e melhorar os algoritmos para um melhor processamento. Por este fato o período de familiarização com as rotinas ACS, HYDROSCAT e TRIOS, foram concluídas com um pouco de atraso (aproximadamente um mês), pois necessitava de certo conhecimento na sintaxe e ambiente Matlab, para que fosse possível obter algum avanço.

### **4.2. Identificação dos Pontos a serem Desenvolvidos nas Rotinas**

Todos os sistemas, códigos e rotinas quando desenvolvidos podem exigir alguma modificação posterior, seja atualizações decorrentes de novas versões do software primário (Matlab), secundário (Microsoft Windows) ou terciário (atualização do software do equipamento de sensoriamento remoto que retorna uma nova saída de resultado), implementação de nova rotina a pedido do usuário, identificação de uma rotina mais eficiente, entre outros fatores que possam se tornar pertinentes desenvolver. Estes são alguns dos motivos de não concluir 100% das identificações dos pontos a serem desenvolvidos nas rotinas já existentes, se considerar pontos até o presente momento excluindo as situações anteriormente descritas posso considerar que 100% dos pontos a serem desenvolvidos foram identificados.

A padronização de variáveis permite uma fácil leitura e identificação das mesmas durante uma análise tornando menos cansativo, criação de comentários descrevendo as principais funções (senão todas) permite entender previamente sua função e o funcionamento de cada linha, generalização de rotinas é muito utilizado no reaproveitamento de códigos tornando fácil a manutenção e mais “limpo” visualmente.

Figura 4.1 – Script com informações gerais



Um exemplo de script com informações relevantes ao analista no momento de fazer alterações ou manutenção.

Fonte: Produção do autor.

Em alguns momentos o ambiente não permite determinadas manipulações de dados por limitação do sistema e nestes casos se faz necessário criar conversões para superar o obstáculo, mas nem sempre a conversão desenvolvida foi a mais eficaz. Estes são alguns dos principais pontos a serem desenvolvidos.

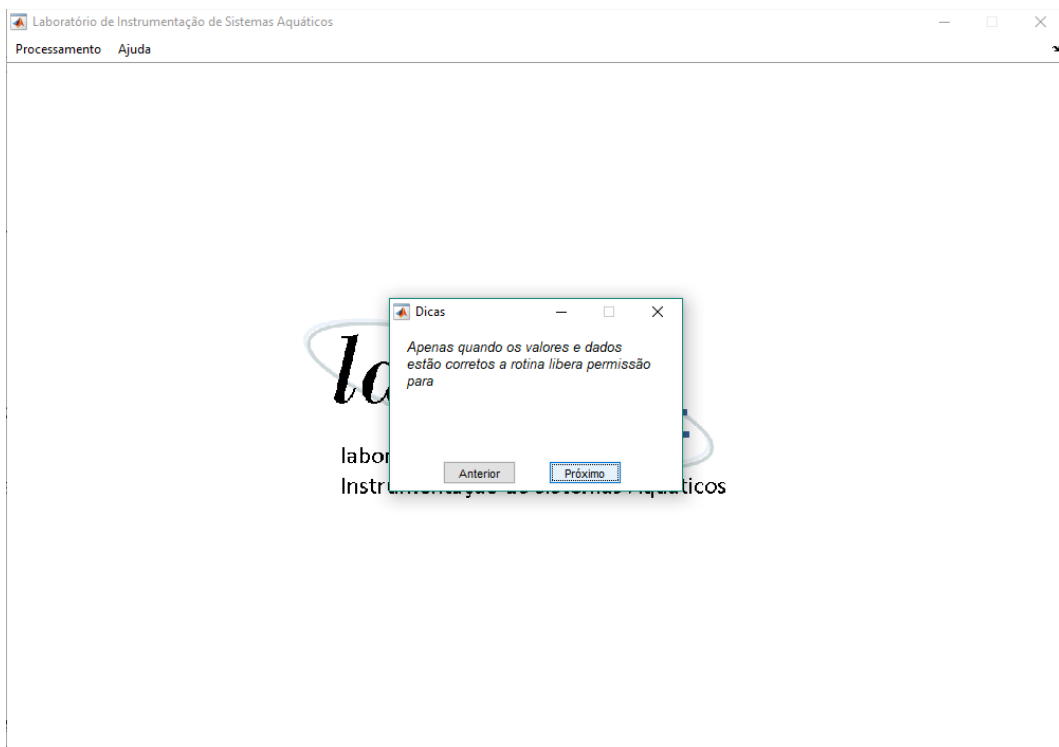
### 4.3. Desenvolvimento das Modificações

Analisando as principais rotinas que necessitavam de modificações foram priorizadas as que são utilizadas com maior frequência ou que exigiam uma demasiada urgência de sua utilização. Apesar de já estarem identificados os pontos de desenvolvimento nas rotinas, é necessário analisar todos os impactos que cada alteração pode causar na execução da rotina, para isso são testadas todas as possibilidades de interferência que possa ocorrer no código desde o início da execução até a saída do resultado.

#### 4.4. Geração de Documentação do Código Desenvolvido

Em paralelo com as modificações nos códigos, está sendo feita a documentação, de uma forma um pouco diferente esta vem se desenvolvendo integralmente com a interface gráfica de cada aplicativo, com o acesso através do menu “ajuda” o usuário pode esclarecer dúvidas quanto a utilização da ferramenta através de um tutorial, outra opção são dicas rápidas para lembrar o usuário de sobre algumas utilidades da ferramenta. Um documento mais técnico também vem se desenvolvendo afim de ser utilizado para eventual manutenção/alteração do código, rotinas, parâmetros, etc.

Figura 4.2 – Interface gráfica com dicas e informações



Fonte: Produção do autor.



## **5 CONCLUSÃO**

O desenvolvimento das tarefas realizadas obteve um resultado muito próximo ao esperado nos objetivos específicos do trabalho. Algumas alterações na interface gráfica foram feitas, apenas detalhes como redução do tamanho da tela, logo do laboratório inserido em espaço em branco.

Como trabalhos ou desenvolvimentos futuros, para uma maior abrangência de análise dos resultados dos equipamentos por usuários que não possuem um computador com o software Matlab, a migração desta tecnologia para ambientes open source como Python ou R.



## REFERÊNCIAS BIBLIOGRÁFICAS

LUTZ, M. (1999). Learning Python. 5. ed. Sebastopol: O'Reilly, 2013. 1540 p.

MATLAB. Programming Fundamentals. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/matlab\\_prog.pdf](https://www.mathworks.com/help/pdf_doc/matlab/matlab_prog.pdf)>. Acesso em: 15 ago. de 2016.

MATLAB. Graphics. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/graphg.pdf](https://www.mathworks.com/help/pdf_doc/matlab/graphg.pdf)>. Acesso em: 19 de out. de 2016.

MATLAB. Function Reference. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/matlab\\_refbook.pdf](https://www.mathworks.com/help/pdf_doc/matlab/matlab_refbook.pdf)>. Acesso em: 10 de dez. de 2016.

MATLAB. Desktop Tools and Development Environment. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/matlab\\_env.pdf](https://www.mathworks.com/help/pdf_doc/matlab/matlab_env.pdf)>. Acesso em: 23 ago. de 2016.

MATLAB. Mathematics. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/math.pdf](https://www.mathworks.com/help/pdf_doc/matlab/math.pdf)>. Acesso em: 10 de dez. de 2016.

PYTHON SOFTWARE FOUNDATION. The Python Standard Library. Disponível em: <<https://docs.python.org/3.6/library/index.html>>. Acesso em: 16 de jan. de 2017.

PYTHON SOFTWARE FOUNDATION. Graphical User Interfaces with Tk. Disponível em: <<https://docs.python.org/2/library/tk.html>>. Acesso em: 20 de fev. de 2017.

MATLAB. MAT-File Format. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/matfile\\_format.pdf](https://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf)>. Acesso em: 23 de mai. de 2017.

MATLAB. Data Import and Export. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/import\\_export.pdf](https://www.mathworks.com/help/pdf_doc/matlab/import_export.pdf)>. Acesso em: 27 de fev. de 2017.

MATLAB. Primer. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/getstart.pdf](https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf)>. Acesso em: 13 de set. de 2016.

MATLAB. Data Analysis. Disponível em: <[https://www.mathworks.com/help/pdf\\_doc/matlab/data\\_analysis.pdf](https://www.mathworks.com/help/pdf_doc/matlab/data_analysis.pdf)>. Acesso em: 10 de mar. de 2017.



MATLAB. App Building. Disponível em:  
<[https://www.mathworks.com/help/pdf\\_doc/matlab/buildgui.pdf](https://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf) >.  
Acesso em: 09 de out. de 2016.