

RELATÓRIO ANUAL 1 – BOLSA Pesquisador TI Nível IV- A.

ALEXSANDRO ALVES LUZ

**Desenvolvimento do Sistema Integrado de Previsão e Segurança Portuária
(SIPS): modelo de previsão para monitoramento de terminais portuários
costeiros - Fase 2**

SUMÁRIO

1. INTRODUÇÃO.....	2
2. ATIVIDADES DESENVOLVIDAS.....	3
2.1. Análise do sistema LINUX em funcionamento.....	3
2.2. Documentação do sistema LINUX em funcionamento.....	3
2.3. Diagramas do sistema LINUX em funcionamento.....	4
2.4. Análise do sistema WEB em funcionamento.....	6
2.5. Treinamento programação WEB.....	8
2.6. Treinamento WEB GIS I.....	9
2.7. Treinamento WEB GIS II.....	10
2.8. Treinamento AWS.....	11
3. RESULTADOS OBTIDOS.....	12
3.1. Criação do novo sistema WEB.....	12
3.2. Criação do novo sistema em LINUX.....	15
3.3. Automação em Shell Script.....	16
3.4. Prevenção a falhas e erros.....	27
3.5. Diagramas do novo sistema LINUX.....	29
3.6. Implementação de novo cliente.....	32
3.7. Conclusão do MVP.....	36
3.8. Ganhos na formação profissional.....	36
4. CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS	36
5. REFERÊNCIAS BIBLIOGRÁFICAS	36
ANEXO 1.....	37
ANEXO 2.....	44
ANEXO 3.....	49

1. INTRODUÇÃO

O produto final proposto nesta iniciativa consiste no MVP (Minimum Viable Product – **Mínimo Produto Viável**) do Sistema Integrado de Previsão e Segurança Portuária (SIPS), sistema de previsão em tempo real das condições de movimentação das embarcações a tracadas, que permita o acesso de diferentes equipes de gestão e operação portuárias.

O sistema tem como objetivo solucionar a necessidade imediata de uma ferramenta de planejamento e resposta confiável para as operações portuárias brasileiras em navios de grande porte, integrando conceitos de modelagem numérica em uma interface amigável que permita ao usuário final a tomada de decisão baseada no acesso e compartilhamento de informações.

Como desafios principais para a FASE 2 do PIPE, foram observadas três macroáreas fundamentais:

1. Sistemas confiáveis: os principais fenômenos ambientais devem ser previstos com confiabilidade
2. Pouca interferência humana: uma vez que o sistema esteja pronto, deve ser minimizada a exigência de tarefas manuais em resposta às falhas dos sistemas
3. Alertas precisos ao cliente: os sistemas devem fornecer alertas facilmente visualizáveis ao cliente, e possibilitar a resolução de problemas operacionais.

Conforme indicado no projeto de pesquisa, as atividades principais em desenvolvimento buscam solucionar os desafios supracitados, contemplando a construção de sistema computacional de alto desempenho adaptado para o processamento de modelos numéricos oceânicos, automação de rotinas, programação de rotinas de computação paralela, prevenção a falhas e erros e pronta notificação de eventuais erros.

Neste relatório são apresentadas as atividades realizadas e os principais resultados obtidos no período de **Agosto-2017 a Agosto-2018**.

Esta bolsa de pesquisa abordou principalmente os temas de **análise e desenvolvimento de sistemas em LINUX e a automação desses sistemas e a apresentação dos dados obtidos na WEB**.

2. ATIVIDADES DESENVOLVIDAS

Análise do sistema LINUX em funcionamento

A primeira tarefa que me foi solicitada no início desta bolsa foi a análise detalhada dos sistemas em LINUX em funcionamento no momento em questão (**Agosto-2017**) para fins de documentação geral, mapeamento e diagramas do sistema.

Ao iniciar as análises pude notar que o sistema era extremamente complexo. Através do relatório final do projeto da fase anterior pude receber informações gerais, além dos próprios scripts e rotinas que continham descrições básicas do que faziam.

Iniciei então o árduo processo para tentar entender, em todas as rotinas e scripts que existiam, como o sistema havia sido projetado e como estava em funcionamento, com uma abordagem da visão macro para a visão micro até chegar à análise detalhada de cada script, linha a linha.

O sistema operacional em funcionamento era um LINUX da distribuição GENTOO que rodava em um servidor de nome NAUTILUS na rede interna da SIAA. Os scripts iniciais eram executados pelo serviço CRON do Nautilus duas vezes ao dia, as 08:00h e as 20:00h, todos os dias automaticamente.

Existiam pelo menos 12 scripts principais envolvidos nos processamentos diários de ponta a ponta do processo inteiro, sendo em sua grande maioria escritos na linguagem Shell Script (bash). Todos os scripts geravam arquivos de log.

Neste relatório não entrarei no detalhe específico das funções de cada um desses scripts por fugir ao escopo tratado aqui, mas a seguir poderão ser encontrados os resultados documentados da análise do sistema em funcionamento.

Documentação do sistema LINUX em funcionamento

Uma vez entendido o sistema em funcionamento, como foi desenhado e como estava funcionando, iniciei o processo de documentação propriamente dito.

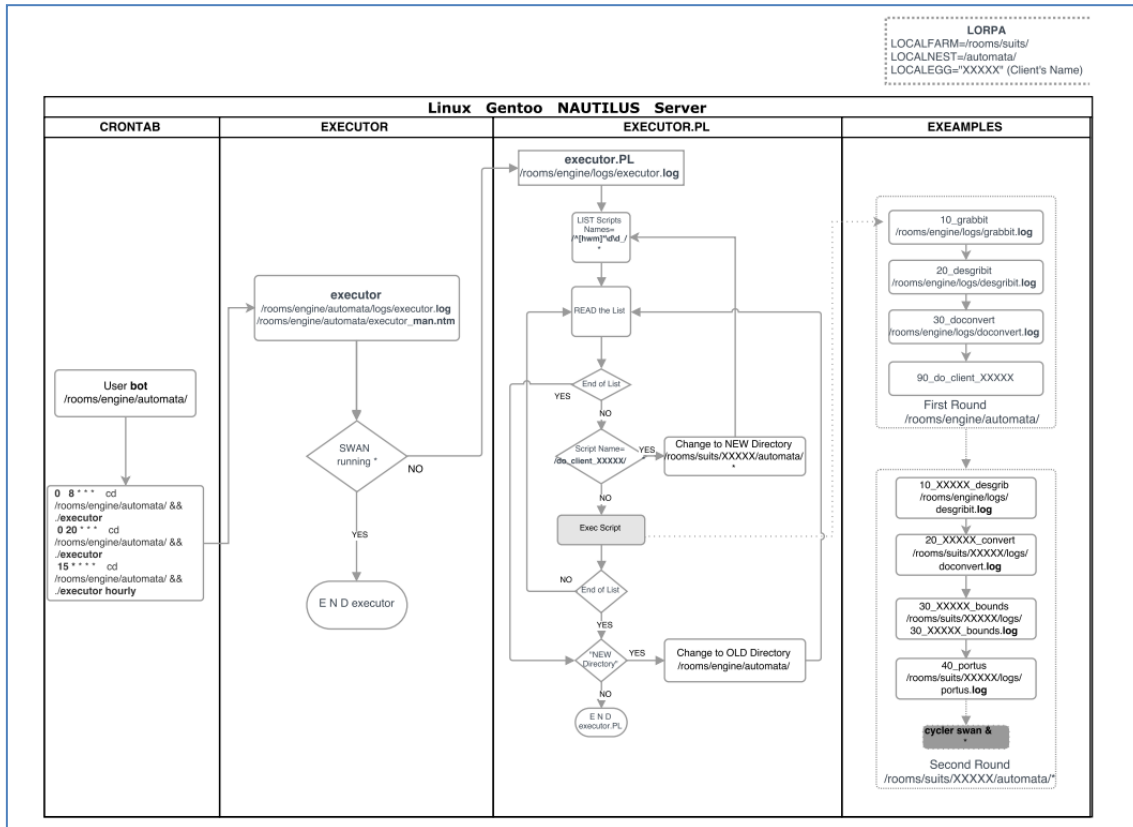
Escolhi usar metodologia simples mas extremamente eficaz ao entendimento, uma documentação na forma de fluxogramas baseados no estilo 'SIPOC', de visão macro mas que remetem aos detalhes, onde cada etapa/função/rotina/script principal é apresentado em colunas verticais paralelas, as quais estão em ordem cronológica do processamento, da esquerda para a direita.

Cada script representado no diagrama mostra o local e nome do seu arquivo de LOG e tem um LINK para o arquivo texto do próprio script, ou seja, para a rotina ou programa escrito nas linguagens Shell Script (bash), Pearl, etc., onde todos os detalhes do script podem ser vistos.

Toda a documentação foi salva em local apropriado e de conhecimento do gestor da SIAA.

Diagrama do sistema LINUX em funcionamento

A seguir o diagrama do sistema em funcionamento em Agosto de 2017 conforme descrito acima.



A seguir imagem com o script inicial inteiro 'executor' escrito na linguagem Shell Script (bash).

```

#!/bin/bash

RC=0;
BOTLAIR="/rooms/engine/automata/";
ROBOT="executor";
ROBOT_CMD="executor.pl";

OPT_SILENT="true"
OPT_LOGFILE="${BOTLAIR}logs/executor.log";

set > /tmp/environment

source ~/.profile
source "${BOTLAIR}lib/dolog.shlib";
source "${BOTLAIR}lib/flags.shlib";

cd "$BOTLAIR";
capture_the_flag "$BOTLAIR" "executor";
RC="$?";
dolog "capture_the_flag said: '$REPLY' with status: '$RC'";
dolog "parametres received, about to pass to the bot:";
dolog "$@";
test "$RC" -eq "0" && exec "${BOTLAIR}${ROBOT_CMD}" "$@"
    
```

A seguir imagem com o início do script 'executor.PL' escrito na linguagem PEARL.



```
executor - Notepad
File Edit Format View Help
#!/usr/bin/perl

use strict;
use IO::File;

require "time.pllib";

my $false = undef;
my $true = 1;
my $logplace = "/rooms/engine/logs/";
my $logname = "executor.log";
my $CLIENTSHOME = $ENV{LOCALFARM};
my $BOOTLAIR = "/rooms/engine/automata/";
my $LOGHANDLE;

main( $ARGV[0], $ENV{PWD} );
exit 0;

sub main
{
    my $hl;
    my $idx;
    my @list;
    my @fpbots;
    my @bots;
    my $item;
    my $ftmp = `tempfile`;
    my $logfile;
    my $client;
    my ( $kind, $place ) = @_ ;

    $place = $BOOTLAIR if not defined $place;
    open $hl, ">>${place}/../logs/${logname}";
    $LOGHANDLE = $hl;
    dolog( "main received: arg1: '$kind' and arg2: '$place'." );
    locatefiles( $place, \@list, \@fpbots, $kind );

    chdir $place;
    dolog( "the following bot(s) was(ere) found: " );
    @bots = sort @list;
```

Após todo o processo de análise e documentação detalhadas do sistema extremamente complexo em funcionamento, em dezembro de 2017 os resultados obtidos foram apresentados ao time da SIAA que os aprovou.

No **ANEXO 1** desse relatório encontra-se o script acima 'executor.PL' inteiro, bem como outros scripts inteiros do sistema em funcionamento em Agosto de 2017.

Análise dos sites WEB em funcionamento

Fazia também parte do projeto em 2017 apresentar os dados obtidos na Internet em um sistema WEB de sites da SIAA onde os futuros clientes pudessem acessar facilmente os resultados das simulações e previsões.

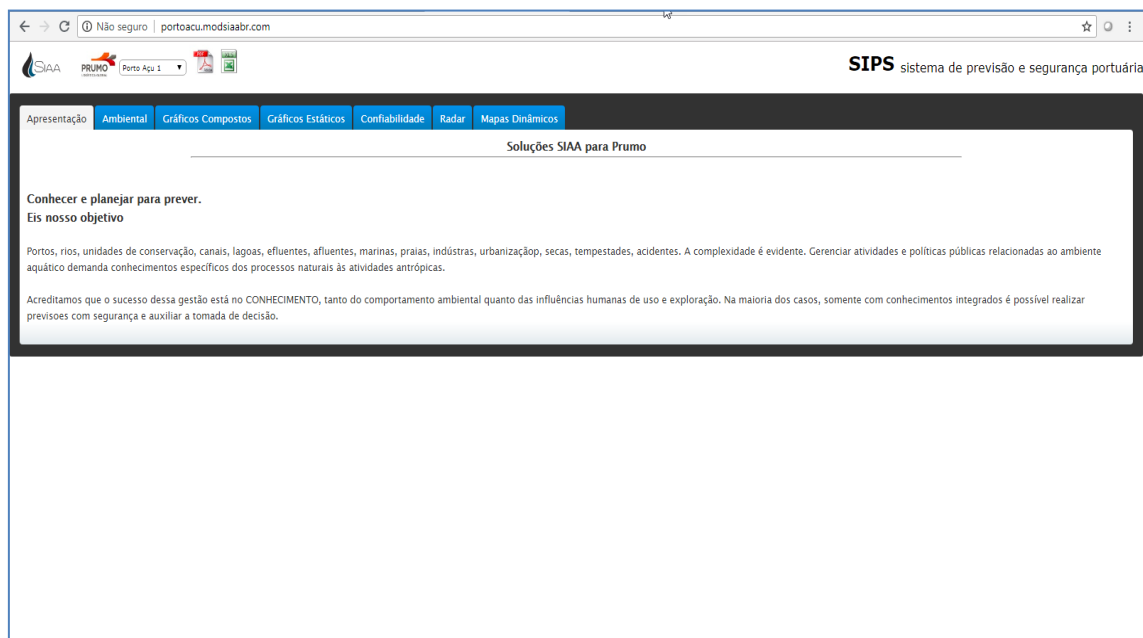
Esse sistema WEB estava hospedado em uma máquina virtual LINUX na AWS, na nuvem da Amazon, era também extremamente complexo e também não havia documentação disponível, exceto as próprias paginas escritas em códigos HTML, CSS e Java Script, principalmente.

Após o término da documentação descrita nos itens anteriores, a partir de Janeiro de 2018, me foi solicitado que estudasse os sites que haviam sido desenvolvidos até então a fim de propor ferramentas e soluções para que os dados obtidos no ambiente LINUX do projeto pudessem ser transferidos automaticamente e visualizados no ambiente WEB do projeto, sem falhas no processamento e transferência de dados.

Também me foi solicitado que novas funções muito mais avançadas de apresentação de mapas interativos e representações dinâmicas de ventos e correntes marítimas fossem adicionadas ao ambiente WEB do projeto.

Alguns exemplos do que estava em funcionamento no ambiente WEB em 2017 poderão ser encontrados abaixo.

A seguir imagem da tela inicial do ambiente WEB em funcionamento.



A seguir imagem do arquivo 'index.html' da tela inicial do ambiente WEB em funcionamento escrito na linguagem HTML.

```

1 <!DOCTYPE html>
2 <head>
3   <meta charset='utf-8'>
4   <link rel='stylesheet' type='text/css' href='css/tabs.css'>
5   <link rel='stylesheet' type='text/css' href='css/index.css'>
6   <link type='text/css' rel='stylesheet' href='css/jquery-ui-1.8.18.custom.css' />
7   <title>SIAA Assessoria em Ambientes Aquáticos</title>
8 </head>
9
10 <body style='margin-top:0px;'>
11   <table width='100%'><tr width='100%'><td width='100%'>
12     <div class='divImage'><img src='gliphos/siaa.png' width='99' height='54'></div>
13     <!-- <div class='divImage' style='margin-top:10px;'><img src='gliphos/prumo-logo.jpg' width='50' height='35'></div> -->
14     <div class='divSpacer'>&nbsp;</div>
15     <div class='divImage' style='margin-top:20px;'><select<option id='porto_acu_1'>SPRJ1</option></select></div>
16     <div class='divSpacer'>&nbsp;</div>
17     <div class='divImage' style='margin-top:2px;'><a href='files/report.pdf' id='pdfdm' onClick='document.location.href="http://portoacu.modsiaabr.com/files/report.pdf"'><img
18       src='gliphos/pdf-logo.jpg' width='41' height='41'></a></div>
19     <div class='divImage' style='margin-top:2px;'><a href='files/report.xls' id='xlsdm' onClick='document.location.href="http://portoacu.modsiaabr.com/files/report.xls"'><img
20       src='gliphos/xls-logo.png' width='39' height='39'></a></div>
21     <div style='text-align:right;float:right;'><div class='titlefont' style='margin-top:10px;'>SIPS&nbsp;</div><div class='subtitlefont' style='margin-
22       top:20px;'>sistema de previsão e segurança portuária</div></div>
23   </td></tr></table>
24   <div id='wrapper' style='margin-top:0px;'>
25     <div id='tabContainer'>
26       <div id='tabs'>
27         <ul>
28           <li id='tabHeader_1'>Apresentação</li>
29           <li id='tabHeader_2'>MAPAS</li>
30           <li id='tabHeader_3'>TABELAS</li>
31           <li id='tabHeader_4'>GRÁFICOS</li>
32           <li id='tabHeader_5'>RISCO</li>
33           <li id='tabHeader_6'>Gráficos Estáticos</li>
34           <li id='tabHeader_7'>Confiabilidade</li>
35           <li id='tabHeader_8'>Radars</li>
36           <li id='tabHeader_9'>Mapas Dinâmicos</li>
37         </ul>
38       </div>
39       <div id='tabContent'>
40         <div class='tabPage' id='tabpage_1'>
41           <center><h3>Soluções SIAA para SPRJ1</h3><div width='70%'></center>
42
43           <br /><br />
44           <p><h3>Conhecer e planejar para prever. <br />
45           Eis nosso objetivo</h3>
46
47           <br />
48           <p>Portos, rios, unidades de conservação, canais, lagoas, efluentes, afluentes, marinas, praias, indústrias, urbanização, secas, tempestades, acidentes. A complexidade é evidente. Gerenciar atividades e políticas públicas relacionadas ao ambiente aquático demanda conhecimentos específicos dos processos naturais às atividades antrópicas.<br />
49
50           <br />
51           <p>Acreditamos que o sucesso dessa gestão está no CONHECIMENTO, tanto do comportamento ambiental quanto das influências humanas de uso e exploração. Na maioria dos casos, somente com conhecimentos integrados é possível realizar previsões com segurança e auxiliar a tomada de decisão.<br />
52
53   </div>
54 </body>
55 </html>

```

A seguir imagem do início do arquivo 'tabler.html' da tela inicial do ambiente WEB em funcionamento escrito na linguagem HTML.

```

1 <!DOCTYPE html>
2 <meta charset="utf-8" />
3
4 <!--
5   HTMLTable : an html5 engine to display and print a table
6   Copyright (C) 2015, Robertol Parra for SIAA - rlparra@gmail.com
7
8   This program is free software: you can redistribute it and/or modify
9   it under the terms of the GNU General Public License as published by
10  the Free Software Foundation, either version 3 of the License, or
11  (at your option) any later version.
12
13  This program is distributed in the hope that it will be useful,
14  but WITHOUT ANY WARRANTY; without even the implied warranty of
15  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16  GNU General Public License for more details.
17
18  You should have received a copy of the GNU General Public License
19  along with this program. If not, see <http://www.gnu.org/licenses/>.
20 -->
21
22 <meta http-equiv="X-UA-Compatible" content="IE=edge">
23 <meta name="viewport" content="width=device-width, initial-scale=1">
24 <link type="text/css" rel="stylesheet" href="css/accordion.css">
25 <link type="text/css" rel="stylesheet" href="css/tabler.css">
26 <link type="text/css" rel="stylesheet" href="css/menu.css">
27 <script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>
28 <title>Tabler</title>
29
30 <div id="cssmenu" style="visibility:visible;margin:0px;">
31 <ul>
32 <!--<li><a href="#" onclick="togglevis('cssmenu');"><small>V</small></a></li>-->
33 <li><a href="#" onclick="reset();"><span>Reset</span></a></li>
34 <li class="active has-sub"><a href="#"><span>Columnas</span></a>
35 <ul>
36 <li class="has-sub"><a href="#"><table width="100%"><tr>
37 <td align="left"><span>Time</span></td>
38 <td align="right"><span><input type="checkbox" checked onchage="hideunhide(0);">seleccionado</span></td>
39 </tr></table></a>
40
41 <li class="has-sub"><a href="#"><table width="100%"><tr>
42 <td align="left"><span>Hava</span></td>
43 <td align="right"><span><input type="checkbox" checked onchage="hideunhide(1);">seleccionado</span></td>
44 </tr></table></a>
45
46 <li class="has-sub"><a href="#"><table width="100%"><tr>
47 <td align="left"><span>Dmva</span></td>
48 <td align="right"><span><input type="checkbox" checked onchage="hideunhide(2);">seleccionado</span></td>
49 </tr></table></a>
50
51 </ul>

```

Treinamento de programação WEB

Em virtude de minha formação profissional não contemplar até o momento em questão (agosto-2017) conhecimentos plenos em programação WEB, solicitei ao gestor da SIAA que eu pudesse fazer treinamentos nessa área específica para que pudesse executar o que me foi solicitado conforme descrito no item anterior. Este treinamento foi necessário para o entendimento de toda a cadeia de transferência de dados do projeto.

Vale notar que a programação WEB é totalmente diferente e extremamente complexa em relação a outras formas de programação tradicionais as quais eu estava habituado por mais de 20 anos de experiência, como por exemplo minha habilidade principal e para a qual fui contratado, que é a linguagem de programação natural dos ambientes em LINUX, chamada de Script Shell (bash).

Minha solicitação não somente foi aprovada como também incentivada pelo gestor da SIAA.

Iniciei o treinamento online introdutório de programação WEB em HTML/CSS/JAVA, tecnologias básicas e essenciais a quase todos os projetos WEB.



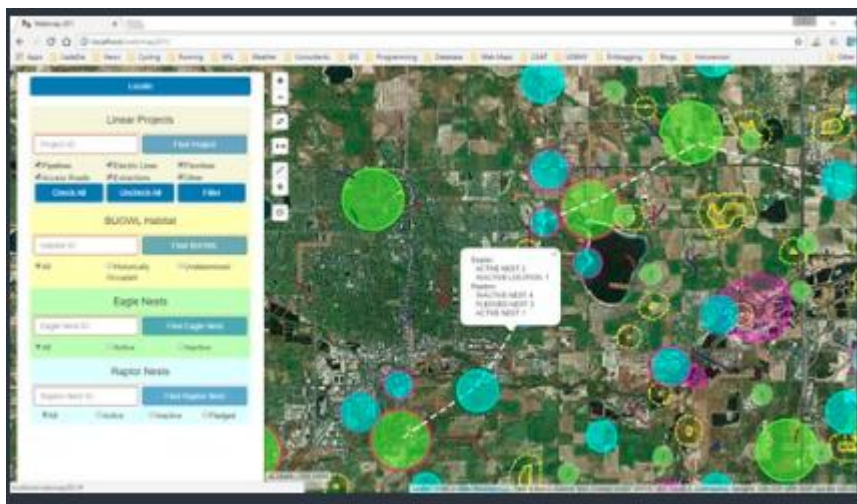
**HTML5, CSS3 e
JavaScript para
Iniciantes**

O curso em questão pode ser encontrado em <https://www.udemy.com/html5-css3-e-javascript-para-iniciantes/?couponCode=BECODE>

Ao concluir esse primeiro treinamento online, apesar da melhora do conhecimento adquirido, pude notar claramente que o curso ainda não era suficiente para enfrentar os desafios propostos.

Treinamento WEB GIS I

Para propor solução adequada, principalmente à solicitação de que mapas interativos e representações dinâmicas de ventos e correntes marítimas fossem adicionados ao ambiente WEB do projeto, novos treinamentos precisariam ser feitos.



Todos os detalhes do curso em questão podem ser encontrados em <https://www.udemy.com/introduction-to-web-programming-for-gis-applications/learn/v4/overview>

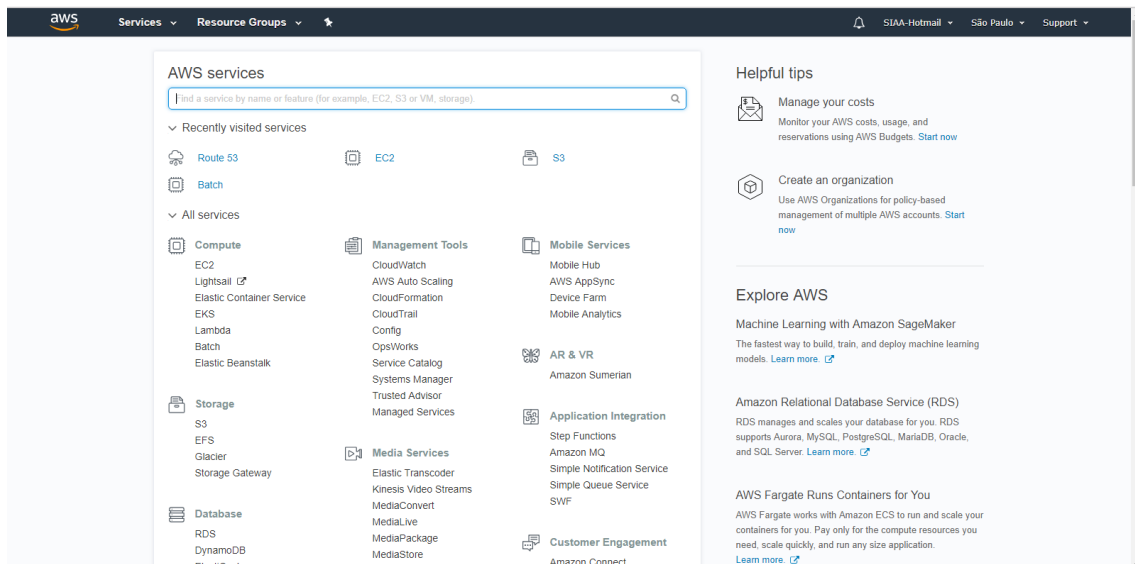
Ao concluir esse terceiro treinamento online, com a melhora muito significativa do conhecimento adquirido, considerei estar minimamente apto a enfrentar os desafios propostos.

Treinamento AWS

Ainda assim faltavam alguns conhecimentos específicos no ambiente da AWS (Amazon Web Services) na nuvem da Amazon, onde estava hospedado o sistema WEB do projeto.

Não havia nenhuma documentação detalhada anterior, exceto o relatório final do projeto da fase anterior que continha apenas informações gerais.

Utilizei então os muitos manuais online da própria AWS para aprender como funcionava todo o esquema em funcionamento da instância LINUX na nuvem, o acesso a essa máquina virtual LINUX, criação de subdomínios, rotas, etc.



Todo o processo de treinamento descrito acima levou aproximadamente dois meses, de janeiro a março de 2018.

3. RESULTADOS OBTIDOS

Criação do projeto para sistema WEB

O novo ambiente WEB do projeto para o MVP foi criado no mesmo ambiente da SIAA na AWS LINUX na nuvem da Amazon, copiando o sistema WEB em funcionamento em 2017, portoacu.modsiaabr.com, para um novo cliente e alterando várias especificações, conforme solicitado pelo gestor da SIAA.

Dessa forma o site do novo cliente ficou em um novo subdomínio do domínio 'modsiaabr.com' da SIAA, a saber, ss.modsiaabr.com, onde novas abas com as novas funcionalidades solicitadas foram adicionadas.

Esse cliente fictício de nome 'SS' refere-se a São Sebastião no litoral do estado de SP.

As duas principais foram as abas 'TABELAS' e 'MAPAS' conforme descrições e imagens abaixo.

Na aba 'TABELAS' ou 'TABLER' foram completamente redimensionadas as colunas e variáveis existentes de acordo com os novos dados recebidos do novo sistema LINUX do projeto.

Nesse momento esses dados já estavam sendo enviados de forma automática pelo sistema LINUX do projeto para o sistema WEB do projeto. Isso será tratado em detalhes posteriormente nesse relatório.

Todas as funcionalidades internas originais da 'tabler' foram mantidas, inclusive o uso do arquivo 'report.js' como base para os dados a serem apresentados.

A seguir imagem da nova 'TABLER'.

Time	Hmo	Tpto	Dato	Vv	Dv	Rq	Temp	Prec	Nuv	Hum
180901	0.0	3.1	0.0	14.0	40.3	18.4	21.8	0.0	31.0	84.7
180904	0.0	1.9	316.3	10.4	38.8	13.7	23.0	0.0	59.1	79.2
180909	0.0	1.9	288.5	7.2	39.9	8.9	23.2	0.0	61.1	81.8
180912	0.0	1.9	281.7	6.8	151.1	8.0	24.1	0.0	55.2	76.3
180915	0.0	1.9	278.8	2.4	87.6	2.1	25.1	0.0	54.8	77.9
180918	0.0	3.0	277.3	12.4	37.9	14.1	25.1	0.0	0.0	72.9
180921	0.0	1.1	274.9	15.0	40.1	19.7	24.7	0.0	59.1	75.4
190900	0.0	4.5	251.8	8.1	52.2	10.9	24.6	0.0	29.0	75.0
190903	0.0	4.5	127.9	5.6	28.4	7.3	23.7	0.0	0.0	78.8
190906	0.0	4.5	124.0	4.3	275.9	3.3	22.6	0.0	0.0	87.3
190909	0.0	3.9	122.5	3.1	224.8	3.8	22.1	0.0	0.0	89.9
190912	0.0	3.9	122.2	6.4	166.2	7.8	22.6	0.0	0.0	87.4
190915	0.0	3.5	121.9	7.8	164.3	9.3	23.3	0.0	46.0	82.8
190918	0.0	3.5	121.9	3.2	67.6	3.9	24.1	0.0	16.7	83.7
190921	0.0	3.1	122.8	9.2	41.3	11.0	23.9	0.0	19.8	81.0
200900	0.0	3.1	123.6	9.0	42.5	11.5	25.5	0.0	54.0	81.8
200903	0.0	3.1	123.7	13.4	42.4	17.6	24.4	0.0	11.1	70.1
200906	0.0	5.7	122.6	18.8	36.3	24.9	24.5	0.0	29.8	69.8
200909	0.0	6.5	122.5	13.7	320.5	17.7	20.4	0.0	100.0	99.0
200912	0.0	6.5	123.0	5.8	272.7	7.9	20.9	0.0	44.4	97.4
200915	0.0	6.5	123.5	11.7	253.3	14.5	21.5	0.0	100.0	98.1
200918	0.0	7.4	127.4	7.7	153.1	10.1	21.4	0.0	100.0	97.4
200921	0.1	7.4	134.2	6.2	111.6	8.0	21.7	0.0	87.7	92.6
210900	0.2	7.4	135.9	5.4	308.5	7.2	21.2	0.0	100.0	98.9
210903	0.3	7.4	136.6	11.6	265.7	14.6	21.0	0.0	100.0	98.6
210906	0.3	7.4	136.1	11.3	250.2	15.0	20.7	0.0	100.0	99.9

A aba 'MAPAS' é uma aba totalmente nova onde a solicitação original, de mapas interativos e representações dinâmicas de ventos e correntes marítimas, foi totalmente atendida.

Foi desenvolvida com a tecnologia LEAFLET – *an open-source JavaScript library for mobile-friendly interactive maps* – <https://leafletjs.com> – e vários de seus plugins.

Todos os botões e informações que aparecem no mapa interativo da aba 'MAPAS' são plugins do Leaflet, a saber.

- Zoomslider
- MousePosition
- MapCenterCoord
- Easy-buttons
- **Velocity**
- **TimeDimension**
- Fullscreen
- Measure
- Scale
- Attribution

Sendo que podemos destacar como os principais plugins dessa aba o 'Velocity' e o 'TimeDimension'.

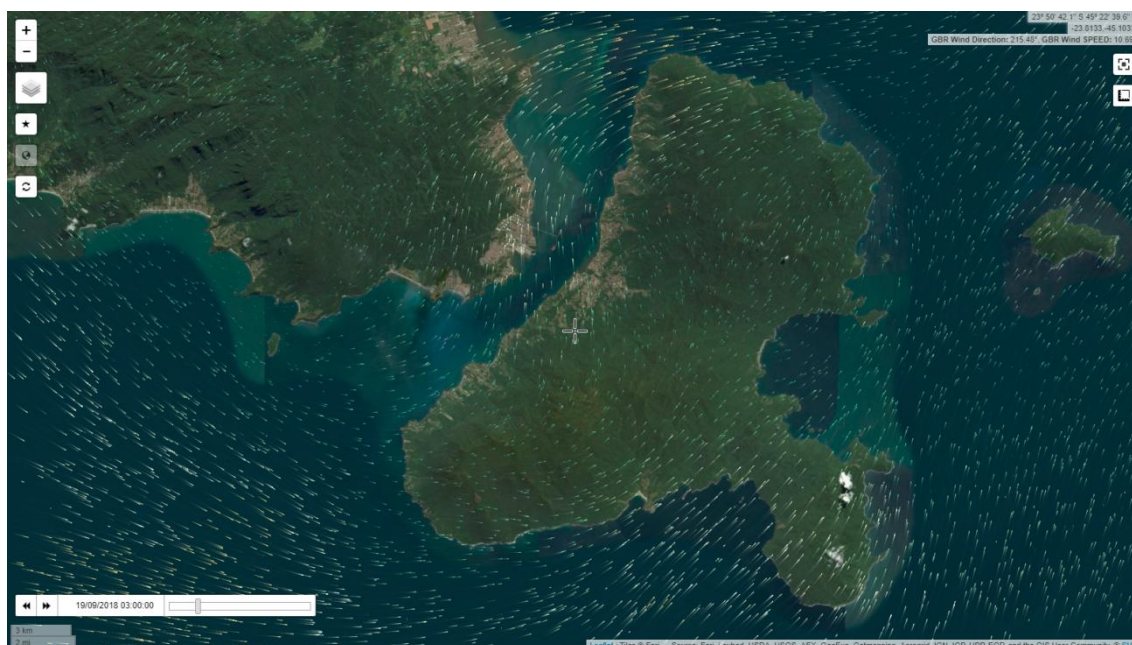
A 'Velocity' é responsável pelas representações ou simulações dinâmicas de VENTOS e ONDAS e todos os detalhes sobre esse plugin podem ser encontrados em <https://github.com/danwild/leaflet-velocity>.

A 'TimeDimension' é responsável por apresentar essas simulações em períodos de tempo, nesse caso de três em três horas, e todos os detalhes sobre esse plugin podem ser encontrados em <https://github.com/socib/Leaflet.TimeDimension>.

A seguir imagem da aba 'MAPAS' com simulação das correntes marítimas (ONDAS).



A seguir imagem da aba 'MAPAS' com simulação atmosférica (VENTOS).



A seguir imagem do início do arquivo ‘MAPS.HTML’ com o código da aba ‘MAPAS’.

```
1 |<!DOCTYPE html>
2
3 |<!-- ***** SIAABR.COM - Alex Luz - 24/07/2018 ***** -->
4
5 |<html lang="en">
6 |<head>
7 |<meta charset="UTF-8">
8 |<title>DEMO HOME SIPS</title>
9 |<link rel="stylesheet" href="src/leaflet.css">
10 |<link rel="stylesheet" href="src/css/bootstrap.css">
11 |<link rel="stylesheet" href="src/plugins/L.Control.ZoomSlider.css">
12 |<link rel="stylesheet" href="src/plugins/L.Control.MousePosition.css">
13 |<link rel="stylesheet" href="src/plugins/L.Control.MapCenterCoord.css">
14 |<link rel="stylesheet" href="src/plugins/leaflet-sidebar.css">
15 |<link rel="stylesheet" href="src/plugins/easy-button.css">
16 |<link rel="stylesheet" href="src/plugins/leaflet-velocity.css">
17 |<link rel="stylesheet" href="src/plugins/leaflet-timedimension.control.css">
18
19 |<link rel="stylesheet" href="https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/leaflet.fullscreen.css"/>
20 |<link rel="stylesheet" href="http://ljagis.github.io/leaflet-measure/leaflet-measure.css" />
21
22 |<script src="src/leaflet-src.js"></script>
23 |<script src="src/jquery-3.2.0.min.js"></script>
24 |<script src="src/plugins/L.Control.ZoomSlider.js"></script>
25 |<script src="src/plugins/L.Control.MousePosition.js"></script>
26 |<script src="src/plugins/L.Control.MapCenterCoord.js"></script>
27 |<script src="src/plugins/leaflet-sidebar.js"></script>
28 |<script src="src/plugins/easy-button.js"></script>
29 |<script src="src/plugins/leaflet-providers.js"></script>
30
31 |<script src="https://api.mapbox.com/mapbox.js/plugins/leaflet-fullscreen/v1.0.1/Leaflet.fullscreen.min.js"></script>
32 |<script src="http://ljagis.github.io/leaflet-measure/leaflet-measure.js"></script>
33
34 |<script src="src/plugins/LeafletPlayback.min.js"></script>
35
36 |<!-- ***** ALTERA PALAVRA 'Speed' PARA 'Height' SOMENTE PARA ONDAS-WAVES - ALEX LUZ - 06/04/2018. -->
37 |<!-- ***** ALTERAÇÃO ESTÁ DENTRO DO CÓDIGO ABAIXO DA VELOCITY - "src/plugins/Leaflet-velocity.js". -->
38 |<script src="src/plugins/Leaflet-velocity.js"></script>
39
40 |<script type="text/javascript" src="https://cdn.rawgit.com/nezasa/iso8601-js-period/master/iso8601.min.js"></script>
41 |<script src="src/plugins/Leaflet.timedimension.js"></script>
42 |<script src="src/plugins/Leaflet.timedimension.util.js"></script>
43 |<script src="src/plugins/Leaflet.timedimension.layer.js"></script>
44 |<script src="src/plugins/Leaflet.timedimension.layer.wms.js"></script>
45 |<script src="src/plugins/Leaflet.timedimension.layer.geojson.js"></script>
46 |<script src="src/plugins/Leaflet.timedimension.player.js"></script>
47 |<script src="src/plugins/Leaflet.timedimension.control.js"></script>
48
49
50 |<style>
51 |#mapdiv {
```

O arquivo ‘MAPS.HTML’ inteiro poderá ser encontrado no **ANEXO 2** desse relatório, juntamente com o arquivo ‘REPORT.JS’ inteiro.

A partir da criação deste projeto, foi possível realizar a integração final do sistema de transferência de dados para web com auxílio de uma empresa especializada.

Criação do novo sistema em LINUX

O novo ambiente LINUX para a fase a qual se refere esse relatório possui DOIS servidores LINUX de nomes PONTOS e NAUTILUS na rede interna da SIAA e também uma máquina virtual LINUX, uma instância Linux AWS na nuvem da Amazon.

O Pontos é um cluster, conjunto de computadores ou unidades de processamento ligados entre si, com 17 dessas unidades interdependentes de processamento, chamadas de nós, sendo 01 nó máster para o controle e 16 nós para o processamento paralelo dos dados.

A distribuição/versão do LINUX instalado atualmente em todos os nós do Pontos é o CENTOS (CentOS Linux release 7.4.1708 - Derived from Red Hat Enterprise Linux 7.4).

A função do Pontos é executar todos os scripts e rotinas relacionados às simulações de VENTOS do projeto SIPS, processando principalmente o WW3 (WaveWatch III Model), e ao final de todo o processamento, transferir os arquivos com os resultados para o Nautilus.

O Nautilus é um servidor comum, e não um cluster, bem menor que o Pontos, mas que tem papel igualmente importante no projeto SIPS, pois sua função é executar todos os scripts e rotinas relacionados às simulações de ONDAS do projeto, processando principalmente o SWAN (Simulating WAVes Nearshore Model), e ao final de todo o processamento, transferir os arquivos com os resultados para a AWS.

A distribuição/versão do LINUX instalado atualmente no Nautilus é o UBUNTU (Ubuntu 16.04.5 LTS).

Os dois servidores se comunicam facilmente através de ferramentas e comandos usuais do LINUX para troca de informações via rede, no caso a rede interna da SIAA.

A Máquina Virtual LINUX, Instância Linux AWS na nuvem da Amazon, tem a função de hospedar os sistemas Web onde são apresentados os resultados das simulações de VENTOS e ONDAS na forma de tabelas e gráficos, ambos dinâmicos, nos sites da SIAA para cada cliente.

A distribuição/versão do LINUX instalado atualmente nessa máquina virtual AWS é o Amazon Linux AMI 2017.09.

Essa Máquina Virtual Linux também se comunica com os outros dois servidores internos da SIAA descritos acima (Pontos e Nautilus) através de ferramentas e comandos usuais do LINUX para troca de informações via rede, nesse caso a Internet.

Automação em Shell Script

Como descrito acima foram utilizados neste projeto, no processo de automação, dois servidores LINUX de nomes Pontos e Nautilus na rede interna da SIAA, sendo que no Pontos foi utilizado o usuário OPERA e no Nautilus o usuário NEMO. Foi utilizada também uma máquina virtual LINUX, uma instância Linux AWS na nuvem da Amazon com o usuário EC2-USER.

Os scripts executores iniciais de todas as rotinas são disparados/iniciados pelo serviço CRON dos dois servidores internos Pontos e Nautilus.

Todos os scripts executores, iniciais ou não, geram LOGS, de nomes e em diretórios específicos, ao serem executados. Um log básico e um log detalhado conforme exemplo abaixo.

<i>Script</i>	<i>Log básico</i>	<i>Log detalhado</i>
<i>executor_opera.sh</i>	<i>executor_opera_log.txt</i>	<i>executor_opera_DETAILS_log.txt</i>

Todos os scripts executores têm tratamento de execução/erro por 'return code' (RC) de todas as rotinas/scripts dentro deles bem como deles próprios.

A cada rotina/script executado seu RC é verificado e uma mensagem de sucesso OU de erro é enviada ao respectivo log.

No Nautilus os scripts executores têm um arquivo de status (.sts) para o controle e tratamento de execução/erro, conforme exemplos abaixo.

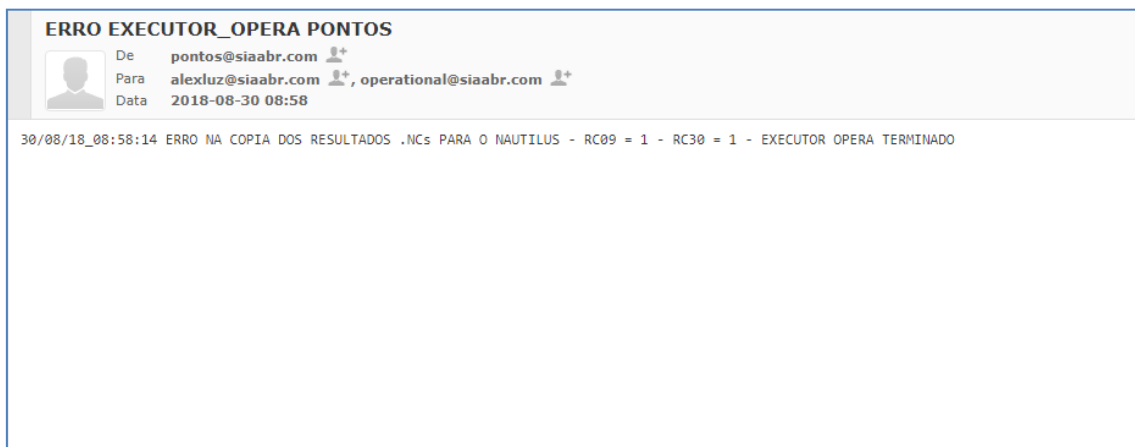
<i>Script</i>	<i>Arquivo de status</i>
<i>exec_nemo.sh</i>	<i>exec_nemo.sts</i>
<i>exec_meteo.sh</i>	<i>exec_meteo.sts</i>
<i>exec_swan.sh</i>	<i>exec_swan.sts</i>

Esses arquivos de status poderão ter somente os três conteúdos abaixo.

VAZIO=Em Processamento 0=OK 1=ERRO

Todos os scripts executores enviam e-mails de notificação em caso de qualquer erro para o e-mail operational@siaabr.com , sendo que para isso utilizam-se os serviços básicos de MAIL previamente instalados em cada servidor.

Seguem abaixo três imagens com três exemplos dos e-mails de notificação de erro.





Também em caso de qualquer erro os scripts executores são cancelados e o processo todo é interrompido.

No **Pontos** todos os scripts ativos ficam no diretório `/home/opera/rooms/engine/automata` . Todos os logs no diretório `/home/opera/rooms/engine/automata/logs` . Todos os scripts inativos e/ou antigos ficam no diretório `/home/opera/rooms/engine/automata/olds` .

O script **executor_opera.sh** é o script executor inicial e está no arquivo 'crontab' do usuário OPERA do Pontos conforme abaixo.

```
#Min Hour Day Month Weekday Command
30 7 * * * /bin/bash
/home/opera/rooms/engine/automata/executor_opera.sh
```

Ele é executado portanto todos os dias as 7:30 da manhã, horário local do servidor Pontos.

No **Nautilus** todos os scripts ativos ficam no diretório `/home/nemo/rooms/engine/automata` . Todos os logs no diretório `/home/nemo/rooms/engine/automata/logs` . Todos os scripts inativos e/ou antigos ficam no diretório `/home/nemo/rooms/engine/automata/olds` .

O script **exec_nemo.sh** é o script executor inicial e está no arquivo 'crontab' do usuário NEMO conforme abaixo.

```
#Min Hour Day Month Weekday Command
30 8 * * * /bin/bash
/home/nemo/rooms/engine/automata/exec_nemo.sh
```

Ele é executado portanto todos os dias as 8:30 da manhã, horário local do servidor Nautilus.

Script executor_opera.sh - PONTOS

Baixa ventos globais da NOAA e executa o WW3.

Abaixo tela com o início do script executor_opera.sh para simples referencia.

Para ver o script real e atualizado acesse o Pontos e edite o
/home/opera/rooms/engine/automata/executor_opera.sh.



```
opera@pontos:~/rooms/engine/automata
/bin/bash
#####
# NEW EXECUTOR_OPERA - SIPS - SIAA #
# Felipe Chagas e Alex Luz - 22/08/2018 #
#####
# Executa '.cronfile' para carregar variaveis NETCDF para o WW3
#HOME/.cronfile > /dev/null 2>&1
#
# Diretorio Home do user OPERA e arquivo de LOG
OPERA=/home/opera
LOG=/OPERA/rooms/engine/automata/logs/executor_opera_log.txt
LOGD=/OPERA/rooms/engine/automata/logs/executor_opera_DETAILS_log.txt
echo "-----" > $LOGD ##### Limpa log DETAILS a cada execucao do executor #####
#
reporter() ##### Function to report errors via MAIL ##
{
MSG="$1"
echo "$MSG" | mail -x pontos@siaabr.com -v -s "ERRO EXECUTOR_OPERA PONTOS" operational@siaabr.com alexluz@siaabr.com >> $LOGD 2>&1
}
#
echo "-----" > $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor OPERA" >> $LOG
### Folder com rotinas e softwares de automatizacao
AUTOM=/OPERA/rooms/engine/automata >> $LOGD 2>&1
#
### Data atual do sistema
ANO=$(date +%Y)
MES=$(date +%m)
DIA=$(date +%d)
##### 1.0 Nivel Global
FOLGLOB=/OPERA/rooms/suits/lglobal >> $LOGD 2>&1
#
### Cria arquivo wget_GFS_global para baixar VENTOS globais ###
octave --no-gui --silent $FOLGLOB/automata/cria_wget_GFS_global.m >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Criado arquivo wget_GFS_global com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA CRIACAO DO ARQUIVO wget_gfs_GLOBAL - EXECUTOR OPERA TERMINADO" >> $LOG
MSG="tail -1 $LOG"; reporter "$MSG"
exit 1
fi
#
#
#
"executor_opera.sh" 201 linhas --08--
```

O arquivo inteiro 'EXECUTOR_OPERA.SH' também pode ser encontrado no **ANEXO 3** deste relatório.

As seguintes rotinas/scripts são processados nessa ordem.

1. Cria arquivo wget_GFS_global para baixar VENTOS globais da NOAA
2. Baixa VENTOS globais
3. Chama WINDER2 para converter VENTOS para formato WW3
4. Copia o arquivo com os componentes U e V gerado pelo WINDER2 para modelo WW3 processar
5. BACKUP do arquivos UV de VENTOS
6. Executa o TESTNODES para checar os nodes antes do WW3
7. R o d a W W 3
8. SE O WW3 DER ERRO EXECUTA TESTNODES NOVAMENTE E WW3 NOVAMENTE
9. Copia resultados de WW3 para o Nautilus via SCP

Script winder2 - PONTOS

Converte arquivo de ventos globais baixado da NOAA para o formato WW3.

Abaixo tela com o início do script winder2 para simples referencia.

Para ver o script real e atualizado acesse o Pontos e edite o /home/opera/rooms/engine/automata/ winder2.



```
#!/bin/bash
INPUTS="$1";
VARIABLE="$2";
OUTPUT="$3";

DESGRIB=/bin/wgrib2;
OPT_OUTPUT_BASE="/home/opera/rooms/suits/iglobal/data/latest_gfs/V";
OPT_DATA_FIELDS="16";

function checkoutput
{
    local TEMP BASE EXT;

    test -z "$OUTPUT" && OUTPUT="/home/opera/rooms/suits/iglobal/data/latest_gfs/latest_gfs.dat";
    if test -f "$OUTPUT"; then
        echo "output file '$OUTPUT' already exist."
        echo "it will be OVERWRITTEN."
        echo "if you reply 'N' to the following question"
        echo "it will be renamed."
        read -p "output file '$OUTPUT' will be overwritten, is it ok (y/N) REPLY;
        if test "$REPLY" != "y" -a "$REPLY" != "Y"; then
            TEMP="date +%Ym%d-%H%M%S";
            BASE="${OUTPUT%.*}";
            EXT="${OUTPUT##*.}";
            test -z "$EXT" && EXT=".${EXT}";
            command mv "$OUTPUT" "${BASE}.${TEMP}.${EXT}";
        fi
    fi
}

function checkproceed
{
    local INPUT FILE VAR ifs;

    echo "" > $OUTPUT;
    echo "-----"
    echo "Input files:"
    echo "-----"
    for FILE in $INPUTS; do echo "input: $FILE"; done
    echo "-----"
    echo "Variables:"
    echo "-----"
    ifs=$IFS; IFS=','; for VAR in $VARIABLES; do echo "variable: $VAR"; done; IFS=$ifs;
    echo "-----"
    read -p "proceed (y/N): " INPUT;
    test "$INPUT" != "y" -a "$INPUT" != "Y" && exit;
}

function checkinput
{
}

"winder2" 121 linhas --0%--
```

As seguintes FUNÇÕES são processadas nessa ordem.

1. checkinput – Checa os valores de entrada (sintaxe)
2. checkoutput – Checa valor de saída (sintaxe)
3. checkproceed – Display dos dados a serem processados
4. Dothepoha – Faz a conversão/formatação dos arquivos

Script testnodes - PONTOS

Testa os nodes do Pontos antes de rodar o WW3. Recria o machinefile conforme os nodes ativos e inativos.

Abaixo tela com o início do script testnodes para simples referencia.

Para ver o script real e atualizado acesse o Pontos e edite o /home/opera/rooms/engine/automata/ testnodes .

```
opera@pontos:~/rooms/engine/automata
#!/bin/bash
#####
# TESTA OS NODES DO PONTOS ANTES DE RODAR WW3
# RECREIA O MACHINEFILE CONFORME OS NODES ATIVOS E INATIVOS
# Alex Luz SIAA 06/08/2018
#####
#
echo "$(date +%d/%m/%y_%H:%M:%S) INICIO do testnodes"
echo
#
MF=/home/opera/demo/ww3/machinefile
PW=/home/opera/demo/ww3/procs_ww3
#
cp $MF ${MF}_OLD; cat $MF; echo
#
### Pega os nodes do /etc/hosts exceto o node00 e joga no array NODES
NODES=($(egrep -i node /etc/hosts | egrep -v node00 | awk {'print $2'}))
echo ${NODES[*]}; echo ${#NODES[*]}; echo
#
declare -i TOT=0 TNA=0 TNI=0 TPA=0 DIF=0 COM=0
declare -i MIN=4 MAX=8; TOT=${#NODES[*]}          ##### Define MIN e MAX e TOT nodes ativos #####
for NOD in ${NODES[*]}
do
    declare -i ST$NOD=1
    ping -c1 $NOD | egrep -i Unreachable > /dev/null 2>&1 && let ST$NOD=1 || let ST$NOD=0          ##### Status do ping de cada node = 0 (OK) ou 1 (NOK - Unreachable)
    echo "$NOD" = $(eval echo \${ST$NOD})
    TNI=$((TNI+$(eval echo \${ST$NOD}))
done
#
TNA=$((TOT - TNI))
#
echo
echo "Total de nodes ATIVOS =" $TNA
echo "Total de nodes INATIVOS =" $TNI
echo
#
[ $TNA -lt $MIN ] && { echo "ATENCAO! TOTAL NODES ATIVOS MENOR QUE $MIN. ***TESTNODES TERMINADO COM RC=1***"; echo; exit 1; }
#
if [ $TNA -gt $MAX ]
then
    echo "ATENCAO! TOTAL NODES ATIVOS (TNA) MAIOR QUE $MAX. TESTNODES CONTINUA COM RC=0 ***MAS ALTERA TNA PARA $MAX***"
#
    DIF=$((TNA-$MAX))
    TNA=$MAX
    TNI=$((TNI+$DIF))
#
    echo
    echo "NOVO total de nodes ATIVOS =" $TNA
*testnodes" 83L, 2245C
```

O arquivo inteiro 'TESTNODES' também pode ser encontrado no **ANEXO 3** deste relatório.

Basicamente faz o seguinte.

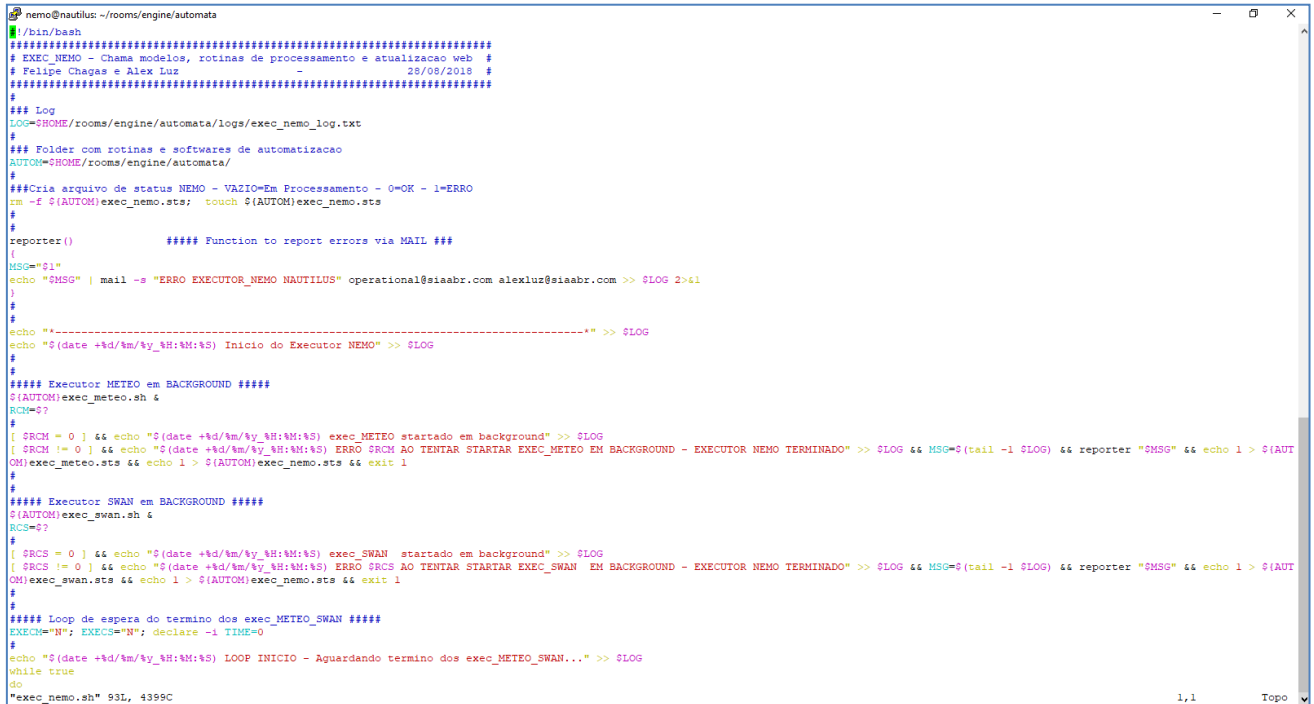
1. Pega os nodes do /etc/hosts exceto o node00 e joga no array NODES
2. Define MIN (4) e MAX (8) e TOT (16) nodes ativos
3. Status do ping de cada node = 0 (OK) ou 1 (NOK - Unreachable)
4. Mostra total de nodes ativos e inativos
5. Se total de nodes ativos (TNA) for MENOR que MIN (4) termina o script com RC=1 (Erro)
6. Se total de nodes ativos (TNA) for MAIOR que MAX continua o script mas altera o TNA para MAX
7. Mostra total de processadores ativos (TPA) que é igual ao TNA x 8 (8 processadores por node)
8. Recria o machinefile conforme os status de cada node e conforme a qtde MAX de nodes permitidos
9. Status 0 'node01:8' - Status 1 '#node01:8' – EXEMPLO

Script exec_nemo.sh – NAUTILUS

Chama modelos, rotinas de processamento e atualização web.

Abaixo tela com o início do script exec_nemo.sh para simples referencia.

Para ver o script real e atualizado acesse o Nautilus e edite o /home/nemo/rooms/engine/automata/ exec_nemo.sh .



```
nemo@nautilus: ~/rooms/engine/automata
/bin/bash
#####
# EXEC_NEMO - Chama modelos, rotinas de processamento e atualizacao web #
# Felipe Chagas e Alex Luz - 28/08/2018 #
#####
### Log
LOG=$HOME/rooms/engine/automata/logs/exec_nemo_log.txt
### Folder com rotinas e softwares de automatizacao
AUTOM=$HOME/rooms/engine/automata/
###Cria arquivo de status NEMO - VAZIO=Em Processamento - 0=OK - 1=ERRO
rm -f $(AUTOM)exec_nemo.sts; touch $(AUTOM)exec_nemo.sts
#
reporter()          ##### Function to report errors via MAIL ###
{
  MSG="$1"
  echo "$MSG" | mail -s "ERRO EXECUTOR_NEMO NAUTILUS" operational@siaabr.com alexluz@siaabr.com >> $LOG 2>&1
}
#
echo ""-----" >> $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor NEMO" >> $LOG
#
##### Executor METEO em BACKGROUND #####
$(AUTOM)exec_meteo.sh &
RCM=$?
#
[ $RCM = 0 ] && echo "$(date +%d/%m/%y_%H:%M:%S) exec_METEO startado em background" >> $LOG
[ $RCM != 0 ] && echo "$(date +%d/%m/%y_%H:%M:%S) ERRO $RCM AO TENTAR STARTAR EXEC_METEO EM BACKGROUND - EXECUTOR NEMO TERMINADO" >> $LOG && MSG=$(tail -1 $LOG) && reporter "$MSG" && echo 1 > $(AUTOM)exec_meteo.sts && echo 1 > $(AUTOM)exec_nemo.sts && exit 1
#
##### Executor SWAN em BACKGROUND #####
$(AUTOM)exec_swan.sh &
RCS=$?
#
[ $RCS = 0 ] && echo "$(date +%d/%m/%y_%H:%M:%S) exec_SWAN startado em background" >> $LOG
[ $RCS != 0 ] && echo "$(date +%d/%m/%y_%H:%M:%S) ERRO $RCS AO TENTAR STARTAR EXEC_SWAN EM BACKGROUND - EXECUTOR NEMO TERMINADO" >> $LOG && MSG=$(tail -1 $LOG) && reporter "$MSG" && echo 1 > $(AUTOM)exec_swan.sts && echo 1 > $(AUTOM)exec_nemo.sts && exit 1
#
##### Loop de espera do termino dos exec_METEO_SWAN #####
EXECN="N"; EXECNS="N"; declare -i TIME=0
echo "$(date +%d/%m/%y_%H:%M:%S) LOOP INICIO - Aguardando termino dos exec_METEO_SWAN..." >> $LOG
while true
do
  "exec_nemo.sh" 93L, 4399C
done
```

As seguintes rotinas/scripts são processados nessa ordem.

1. Exec_METEO em BACKGROUND
2. Exec_SWAN em BACKGROUND
3. Loop de espera do termino dos exec_METEO_SWAN
4. Limite de espera de 240 minutos (4 horas) se ultrapassado termina o exec_nemo.sh com RC=1 (Erro).
5. Exec_CLIENT

Script exec_meteo.sh - NAUTILUS

Rodadas operacionais de WRF (VENTOS) - Origem: Belgingur.

Abaixo tela com o início do script exec_meteo.sh para simples referencia.

Para ver o script real e atualizado acesse o Nautilus e edite o /home/nemo/rooms/engine/automata/ exec_meteo.sh .

```
nemo@nautilus: ~/rooms/engine/automata
/bin/bash
#####
# EXEC_METEO - Rodadas operacionais de WRF - Origem: Belgingur #
# Felipe Chagas e Alex Luz - 27/08/2018 #
#####
### Logs
LOG=$HOME/rooms/engine/automata/logs/exec_meteo_log.txt
LOGD=$HOME/rooms/engine/automata/logs/exec_meteo_DETAILS_log.txt
#
### Folder com rotinas e softwares de automatizacao
AUTOM=$HOME/rooms/engine/automata/
#
### Cria arquivo de status - VAZIO=Em Processamento - 0=OK - 1=ERRO
rm -f ${AUTOM}exec_meteo.sts: touch ${AUTOM}exec_meteo.sts
#
#
reporter() ##### Function to report errors via MAIL ###
{
MSG="01"
echo "$MSG" | mail -s "ERRO EXECUTOR_METEO NAUTILUS" operational@siaabr.com alexluz@siaabr.com >> $LOG 2>&1
}
#
### Limpa arquivo de log DETAILS sempre no inicio do executor
echo "-----" > $LOGD
echo "-----" > $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor METEO" >> $LOG
#
#####
# Rodadas de modelos numericos #
#####
#### 1.Nivel Global
#
#### 1.1 Nivel Atlantico Sul
FOLATLAS=$HOME/rooms/suits/iglobal/latlas >> $LOGD 2>&1
#
#####while true ##### Loop para o caso de o arquivo nao estar disponivel ainda ##### INCLUIR LIMITE DE ESPERA ATE AS 18:00 ?????? #####
#####do
#####set -C https://siaa.belgingur.is/wrfout/wrfout_siaa_01_12.5km-$(date +%Y-%m-%d)T00:00.nc -P $FOLATLAS/data/latest_wrf --http-user=siaa --http-passwd=9HBbrV91YAM0418v >> $LOGD 2>&1
#####
#####egrep -i "404 Not Found" $LOGD >> $LOG && { echo "$(date +%d/%m/%y_%H:%M:%S) FILE NOT FOUND... AGUARDA 10 MIN E TENTA NOVAMENTE..." >> $LOG; mv $LOGD $(LOGD)_404_FNF ; sleep 10m; } || break
#####
#####done
#####
#####echo "$(date +%d/%m/%y_%H:%M:%S) Baixado Grade Atlas WRF" >> $LOG
#
declare -i TIME=0
#
"exec_meteo.sh" 123L, 6013C
1,1 Topo
```

As seguintes rotinas/scripts são processados nessa ordem.

- Rodadas de Modelos Numéricos
 1. Baixa arquivo 'Grade Sprj WRF' de siaa.belgingur.is
 2. Loop para o caso de o arquivo não estar disponível ainda – Aguarda 10 minutos e tenta baixa-lo novamente.
 3. Limite de espera de 360 minutos (6 horas) se ultrapassado termina o exec_meteo.sh com RC=1 (Erro).
- Pós Processamento de Resultados de Modelos
 1. WRF Pos Proc 001public (retirado de grade Atlas)
 2. WRF Pos Proc 002ss (retirado de grade Atlas)
 3. WRF Pos Proc 003demo (retirado de grade Atlas)

Script exec_swan.sh - NAUTILUS

Rodadas operacionais do modelo SWAN (ONDAS).

Abaixo tela com o início do script exec_swan.sh para simples referencia.

Para ver o script real e atualizado acesse o Nautilus e edite o
/home/nemo/rooms/engine/automata/ exec_swan.sh .

```
nemo@nautilus: ~/rooms/engine/automata
/bin/bash
#####
# EXEC_SWAN - Rodadas operacionais do modelo SWAN - SIPS/SIAA #
# Felipe Chagas e Alex Luz - 27/08/2018 #
#####
### Logs
LOG=$HOME/rooms/engine/automata/logs/exec_swan_log.txt
LOGD=$HOME/rooms/engine/automata/logs/exec_swan_DETAILS_log.txt
#
### Folder com rotinas e softwares de automatizacao
AUTOM=$HOME/rooms/engine/automata/ #folder com rotinas e softwares de automatizacao
#
### Cria arquivo de STATUS - Vazio=EM PROCESSAMENTO - 0=OK - 1=ERRO
rm -f ${AUTOM}exec_swan.sts; touch ${AUTOM}exec_swan.sts
#
#
reporter()
{
MSG="01"
echo "$MSG" | mail -s "ERRO EXECUTOR_SWAN NAUTILUS" operational@siaabr.com alexluz@siaabr.com >> $LOG 2>&1
}
#
### Limpa arquivo de log DETAILS sempre no inicio do executor
echo "-----" > $LOGD
echo "-----" > $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor SWAN" >> $LOG
#
echo "$(date +%d/%m/%y_%H:%M:%S) LOOP INICIO - Aguardando arquivos .NC do WW3 no PONTOS" >> $LOG
#
DIR1="/home/nemo/rooms/suite/global/data/latest_vw3/"
DIR2="/home/nemo/rooms/suite/global/atlas/data/latest_vw3/"
FILE1="N"; FILE2="N"; LOOP="N"; declare -i TIME=0
#
rm -f ${DIR1}gridglo30min.nc ${DIR2}gridatl09min.nc
#
while true
do
[ -s ${DIR1}gridglo30min.nc ] && [ "${FILE1}" = "N" ] && echo "$(date +%d/%m/%y_%H:%M:%S) Arquivo gridglo30min.nc encontrado no dir $DIR1 em $TIME minutos" >> $LOG && FILE1="Y"
[ -s ${DIR2}gridatl09min.nc ] && [ "${FILE2}" = "N" ] && echo "$(date +%d/%m/%y_%H:%M:%S) Arquivo gridatl09min.nc encontrado no dir $DIR2 em $TIME minutos" >> $LOG && FILE2="Y"
[ "${FILE1}" = "Y" -a "${FILE2}" = "Y" ] && LOOP="Y" && break
sleep 1m; let TIME=TIME+1
#
done
#
0
#
"exec_swan.sh" 207L, 11782C
1,1 Topo
```

As seguintes rotinas/scripts são processados nessa ordem.

- LOOP INICIO - Aguardando arquivos .NC do WW3 no PONTOS
 1. Limite de espera de 120 minutos (2 horas) se ultrapassado termina o exec_swan.sh com RC=1 (Erro).
- Rodadas de Modelos Numéricos
 1. Usa WW3 Atlas para criar bounds de Swan1 e Swan5 de Sp_rj
 2. Modifica INPUTS para rodar swan SPRJ
 3. Roda SWAN1 e SWAN5
 4. Copia restarts gerados para as simulações do próximo dia
 5. Copia Nest gerado pela grade Sprj
 6. Modifica INPUT para rodar Swan SS
 7. Roda SWAN-NEST-SS
 8. Copia restarts gerados para as simulações do próximo dia
- Pos Processamento de resultados de modelos
 1. WW3 Pos Proc 001public
 2. SWAN Pos Proc 002ss
 3. SWAN Pos Proc 003demo

Script exec_client.sh - NAUTILUS

Converte os arquivos CSVs WebPoints para os 'reportS.js' para as 'TABLERs' dos sites XXXX.modsiaabr.com que ficam na AWS.

O script já está preparado para fazer todo o processo de conversão listado mais abaixo para todos os clientes que existirem.

O diretório base é o /home/nemo/rooms/steaks/ e deverá conter necessariamente a estrutura de subdiretórios como a listada abaixo.

```
nemo@nautilus: ~/rooms/steaks/001public
nemo@nautilus:~/rooms/steaks$ ls -l
total 16
drwxr-xr-x 8 nemo nemo 4096 Ago 23 11:39 001public
drwxr-xr-x 8 nemo nemo 4096 Jun 25 20:34 002ss
drwxrwxr-x 8 nemo nemo 4096 Jun 26 10:49 003demo
drwxrwxr-x 2 nemo nemo 4096 Jun 26 10:47 004sp
nemo@nautilus:~/rooms/steaks$ cd 001public/
nemo@nautilus:~/rooms/steaks/001public$ ls -l
total 24
drwxr-xr-x 2 nemo nemo 4096 Ago 24 18:17 webgrids_swan
drwxr-xr-x 2 nemo nemo 4096 Ago 24 14:56 webgrids_wrf
drwxr-xr-x 2 nemo nemo 4096 Ago 27 18:49 webpoints_swan
drwxr-xr-x 2 nemo nemo 4096 Ago 27 18:50 webpoints_wrf
drwxr-xr-x 2 nemo nemo 4096 Ago 23 11:39 webrisks
drwxr-xr-x 2 nemo nemo 4096 Set 5 11:09 webtabs
nemo@nautilus:~/rooms/steaks/001public$
```

Abaixo o arquivo 'report.js' MODELO em /home/nemo/rooms/engine/automata/report.js MODELO de onde é extraída a formatação das 'TABLERs' que serão exibidas nos sites dos clientes na AWS.

[illegible]

Abaixo tela com o início do script `exec client.sh` para simples referencia.

Para ver o script real e atualizado acesse o Nautilus e edite o `/home/nemo/rooms/engine/automata/exec client.sh`.

```

nemo@nautilus: ~/rooms/engine/automata
# /bin/bash
#####
# Converte os CSV WebPoints para os 'reportS.js' para as 'TABLERs' dos sites XXXX.modsiaabr.com #
# Alex Luz para SIAA - 24/09/2018 #
#####

reporter()
{
    MSG="01"
    echo "$MSG" | mail -s "ERRO EXECUTOR_CLIENT NAUTILUS" operational@siaabr.com alexluz@siaabr.com ##### TIRAR O MEU E-MAIL... #####
}

echo "-----"
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor CLIENT"
#
REAL_HOME=$HOME
AUTOM=$(REAL_HOME)/rooms/engine/automata/"
BASE_DIR=$(REAL_HOME)/rooms/steaks/"
REPORT_JS_BASE=$(AUTOM/report.js_MODELO";
QWPS=0; QWFW=0; ACLIS=(); I=0;
#
check()
{
    ##### Checagem geral #####
    echo "$(date +%d/%m/%y_%H:%M:%S) Inicio da checagem"
    INPUTS=""
    rm -f $(BASE_DIR)WvpS* $(BASE_DIR)WvpW* ##### Arquivos AUXILIARES com NOMES dos WebPoints Swan e WRF #####
    rm -f $(BASE_DIR)LvpS* $(BASE_DIR)LvpW* ##### Arquivos AUXILIARES com LINHAS dos WebPoints Swan e WRF #####
    rm -f $(BASE_DIR)DvpS* $(BASE_DIR)DvpW* ##### Arquivos AUXILIARES com DATAS dos WebPoints Swan e WRF #####
    #
    echo $BASE_DIR BASEDIR
    [ $(BASE_DIR)/* = "$(BASE_DIR)/*" ] 2> /dev/null && MSG="BASE_DIR VAZIO - EXEC_CLIENT ENCERRADO COM RC=1..." && echo $MSG && reporter "$MSG" && exit 1
    cd $BASE_DIR
    for CLIENT in $(BASE_DIR)*
    do
        [ ! -d "$CLIENT" ] && echo "$CLIENT nao eh diretorio. Desprezado..." && continue
        [ $(CLIENT)/* = "$(CLIENT)/*" ] 2> /dev/null && echo "$CLIENT client vazio. Desprezado..." && continue
        #
        echo $CLIENT CLIENT
        cd $(CLIENT)
        ACLIS[I]=$CLIENT
        I=$((I + 1))
        #
        for DIR in $(CLIENT)/webpoints*
        do
            "exec_client.sh" 299L, 9966C

```

O arquivo inteiro 'EXEC_CLIENT.SH' também pode ser encontrado no **ANEXO 3** deste relatório.

As seguintes FUNÇÕES são processadas nessa ordem.

1. check - Checagem geral dos diretórios e arquivos envolvidos
 - a. Diretórios vazios
 - b. Arquivos soltos no lugar de diretórios
 - c. QTDES de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
 - d. NOMES de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
 - e. LINHAS de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
 - f. DATAS de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
2. join – Junta os CSVs originais e cria novos CSVs para e-mails
 - a. Junta os CSVs originais e os converte (convert function) para os reports.js
 - b. Cria novos CSVs para enviar em anexos aos e-mails aos clientes
3. convert - Converte CSV original em report.js
4. transfer – Transfere os arquivos envolvidos para a AWS (Máquina Linux na Nuvem da Amazon)
 - a. Copia reportS.js e JSONs para AWS (Cliente 001public <=> Pasta 'public' da AWS)
 - b. Copia reportS.js e JSONs para AWS (Cliente 002ss <=> Pasta 'ss' da AWS)
 - c. Copia reportS.js e JSONs para AWS (Cliente 003demo <=> Pasta 'demo' da AWS)
 - d. ...
5. sendcsv - Envia e-mails com novos CSVs anexos para clientes e SIAA conforme linhas exemplos abaixo.

```

#      EMAIL_CLIENT $1      EMAIL_SIAA $2      SUBJECT $3      DIR-
CLI $4 BODY OF EMAIL... $5
sendcsv "operational@siaabr.com" "operational@siaabr.com" "Report do dia $(date +%d/%m/%y) - Canal de São Sebastião"
"002ss" "Reports do dia $(date +%d/%m/%y) em anexo.^M^MVisualize também no sistema web:
http://ss.modsiaabr.com/tabler.html"

```

Essa linha da função 'sendcsv' para envio do e-mail com os CSVs em anexo para os clientes precisa ser adicionada manualmente ao final do script exec_client para cada novo cliente.

Prevenção a falhas e erros

Como o projeto SIPS foi praticamente todo desenvolvido em ambientes LINUX e utilizando rotinas em Shell Script para a automação geral do sistema, foram desenvolvidas e implementadas algumas técnicas de prevenção e notificação de falhas dos processamentos automáticos.

Todas as rotinas do projeto rodam diariamente de forma automática e em background e portanto não podem, e não devem, ter intervenção humana durante os processamentos, justamente para evitar erros e falhas causados por essa intervenção.

Por isso, para que se tenha controle e monitoramento do que, como e quando, está sendo processado, as seguintes metodologias de prevenção e notificação de falhas foram adotadas. São técnicas relativamente simples mas extremamente eficientes e consagradas por muitos anos nos ambientes LINUX.

Todas elas foram testadas exaustivamente com sucesso durante o desenvolvimento do MVP e estão estabelecidas e rodando automaticamente todos os dias.

TODOS os scripts:

1. Geram logs básicos e logs detalhados.
2. Têm tratamento de erros por return code.
3. Gravam mensagens de sucesso ou de erro nos logs.
4. Envia e-mails de notificação em caso de qualquer erro.
5. Cancelam a si próprios em caso de qualquer erro.
6. Interrompem o processo inteiro em caso de qualquer erro.
7. Têm controle de tempo máximo de espera quando aguardam por resultados de outros scripts.

Servidor PONTOS

Script testnodes

Como o servidor PONTOS é um cluster com vários nós, o script TESTNODES foi especialmente desenvolvido para evitar que eventuais falhas nos nós interrompam o processamento da simulação WW3.

Ele testa todos os nós que existirem no Pontos ANTES do início da simulação do WW3 para verificar quais nós estão ativos e quais estão inativos.

Então ele recria o arquivo 'machinefile' conforme os nós ativos e inativos. Esse arquivo será usado pelo WW3 para distribuir o processamento da simulação somente entre os nós ativos.

Existe também um controle das quantidades mínimas e máximas permitidas de nós ativos, sendo que atualmente a quantidade mínima é 4 e a máxima é 8, de um total de 16 nós existentes no Pontos, além do nó máster.

Caso a quantidade total de nós ativos for MENOR que a quantidade mínima permitida o script é cancelado com return code 1 (erro), cancelando também o processo inteiro e enviando emails de notificação de erro.

Caso a quantidade total de nós ativos for MAIOR que a quantidade máxima permitida o script continua mas altera a quantidade total de nós ativos para a quantidade máxima permitida.

Script executor_opera.sh

Uma vez que o testnodes foi executado com sucesso e o WW3 já estiver rodando, um controle foi implementado no script executor_opera.sh, que é o script que chama o script testnodes e o WW3, de forma que se o WW3 der erro DURANTE o processamento da simulação o testnodes será executado novamente conforme descrito acima e se terminar com sucesso o WW3 também será executado novamente.

Isso garante um controle de erros e uma prevenção a falhas extremamente eficientes, pois todos os nós são testados ANTES do processamento do WW3 e, além disso, em caso de erro DURANTE o processamento do WW3, todos os nós são testados novamente e o WW3 é processado mais uma vez.

Optou-se a princípio por deixar esse 'loop' de testes dos nós e processamentos do WW3 em apenas mais UMA tentativa.

Vale notar no entanto que poderiam ser várias tentativas, bastando para isso apenas a implementação apropriada no script executor_opera.sh, bem como levar em consideração o tempo total de processamento dessas várias tentativas para não comprometer o tempo hábil de publicação dos resultados.

Servidor NAUTILUS

Script exec_client.sh

O script exec_client.sh que é o script que finaliza todo o processo, convertendo e formatando os dados gerados pelos scripts anteriores para serem visualizados nos sites dos clientes, foi desenvolvido com vários controles de prevenção a falhas como os listados abaixo.

1. Está preparado para executar todo o processo de conversão e formatação para todos os clientes que existirem no Nautilus automaticamente, e não clientes fixos dentro do script.
2. Utiliza arquivo MODELO para extrair a formatação das tabelas que serão exibidas nos sites, e não valores fixos dentro do script.
3. Executa checagens em todos os diretórios e arquivos envolvidos no processo de conversão/formatação, como as listadas abaixo.
 - a. Diretórios vazios
 - b. Arquivos soltos no lugar de diretórios

- c. QTDES de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
- d. NOMES de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
- e. LINHAS de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF
- f. DATAS de arquivos diferentes nos diretórios WEBPOINTS_SWAN_WRF

Diagramas do novo sistema LINUX.

A automação geral do projeto SIPS foi toda desenvolvida em rotinas Shell Script em ambientes LINUX e já foi devidamente detalhada em outra parte desse relatório.

Para facilitar a compreensão de como, onde e quando essas rotinas são processadas, foram desenhados alguns diagramas para que todo o processo pudesse ser visualizado graficamente na forma de fluxogramas simples.

Esses fluxogramas ou diagramas listados abaixo mostrarão, primeiramente de forma macro e depois de forma detalhada, cada etapa do processo inteiro, na ordem em que ocorrem.

Diagrama 1 – Visão Macro Geral – Pontos - Nautilus

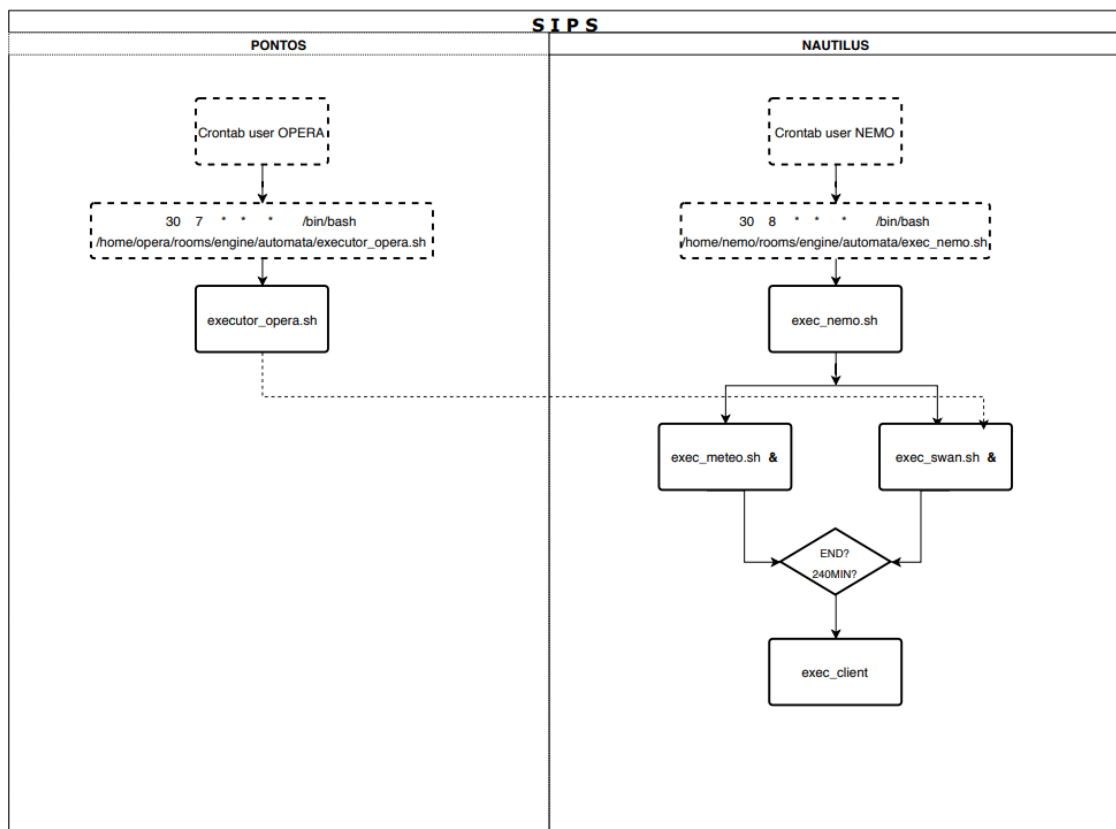


Diagrama 2 – Detalhe Um – Executor_opera.sh no Pontos – Exec_nemo.sh no Nautilus

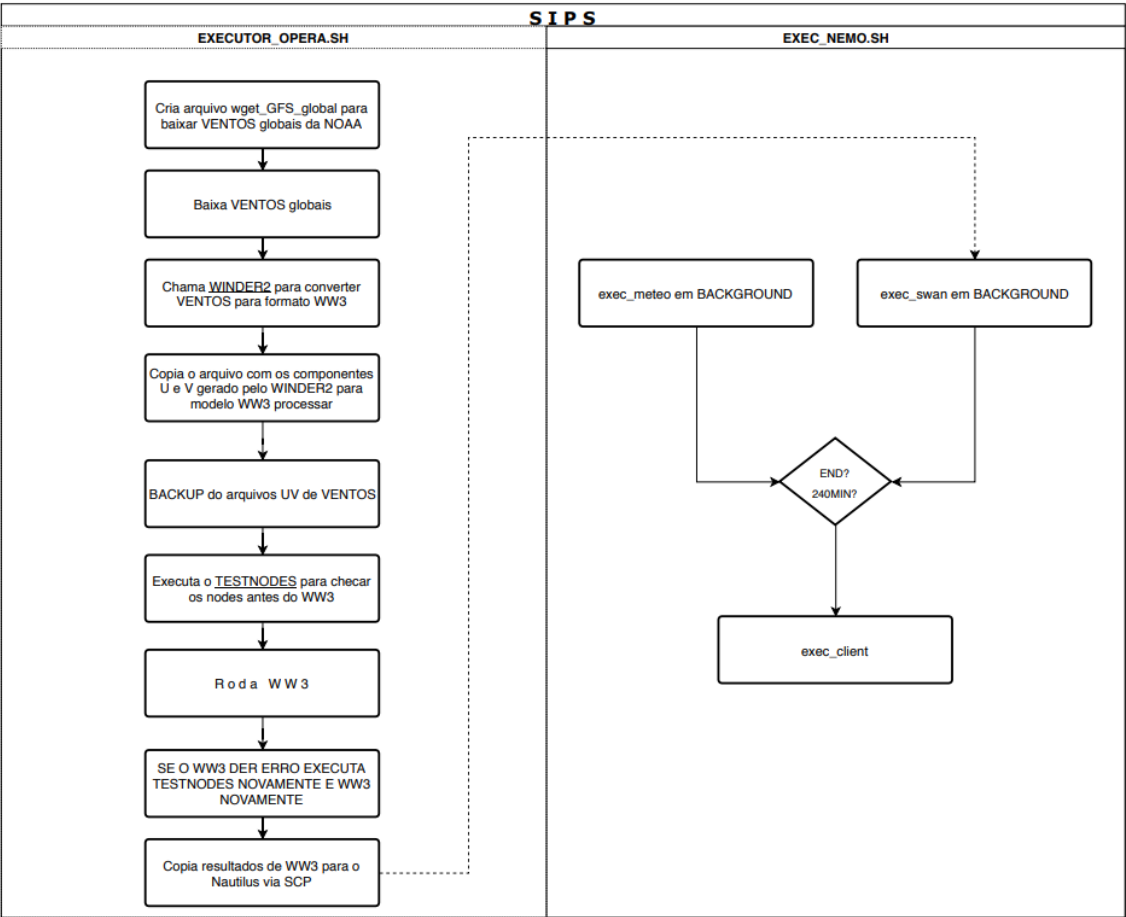


Diagrama 3 – Detalhe Dois – Exec_meteo.sh – Exec_swan.sh - Nautilus

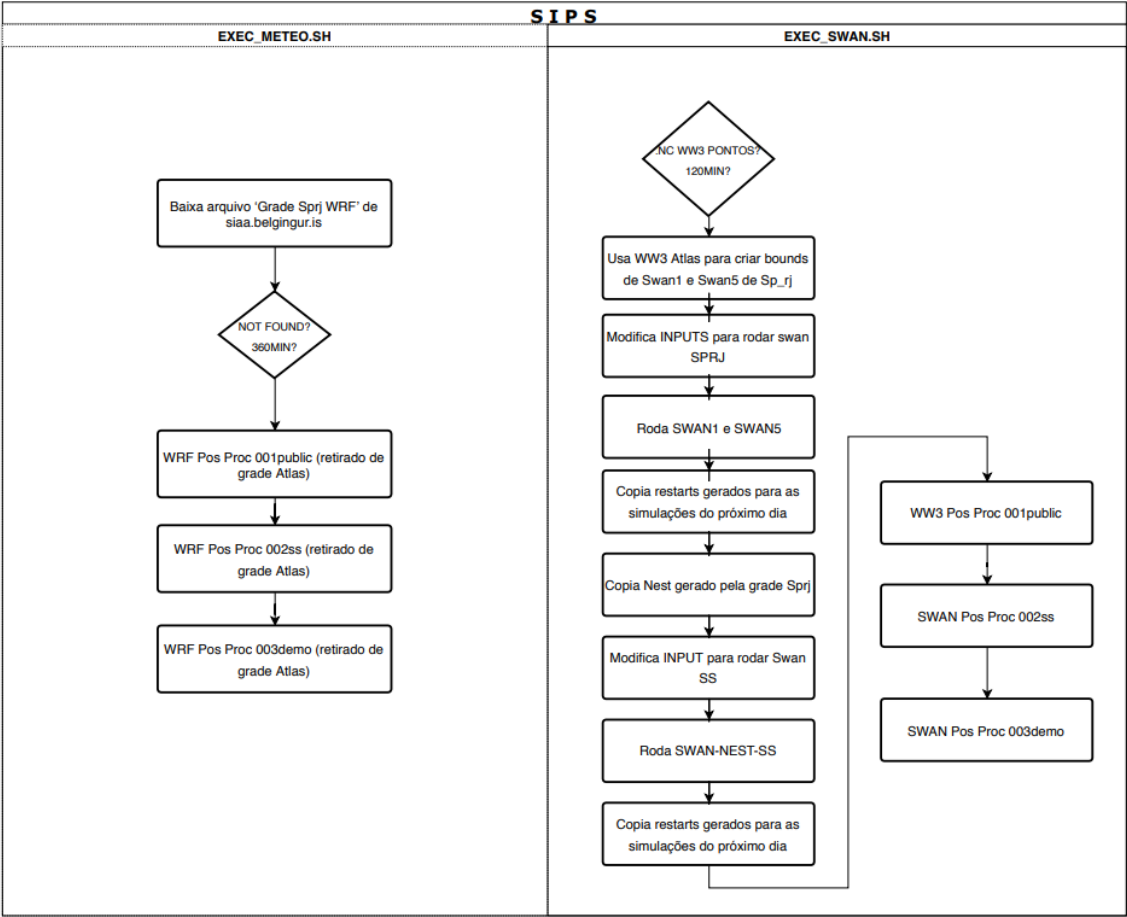
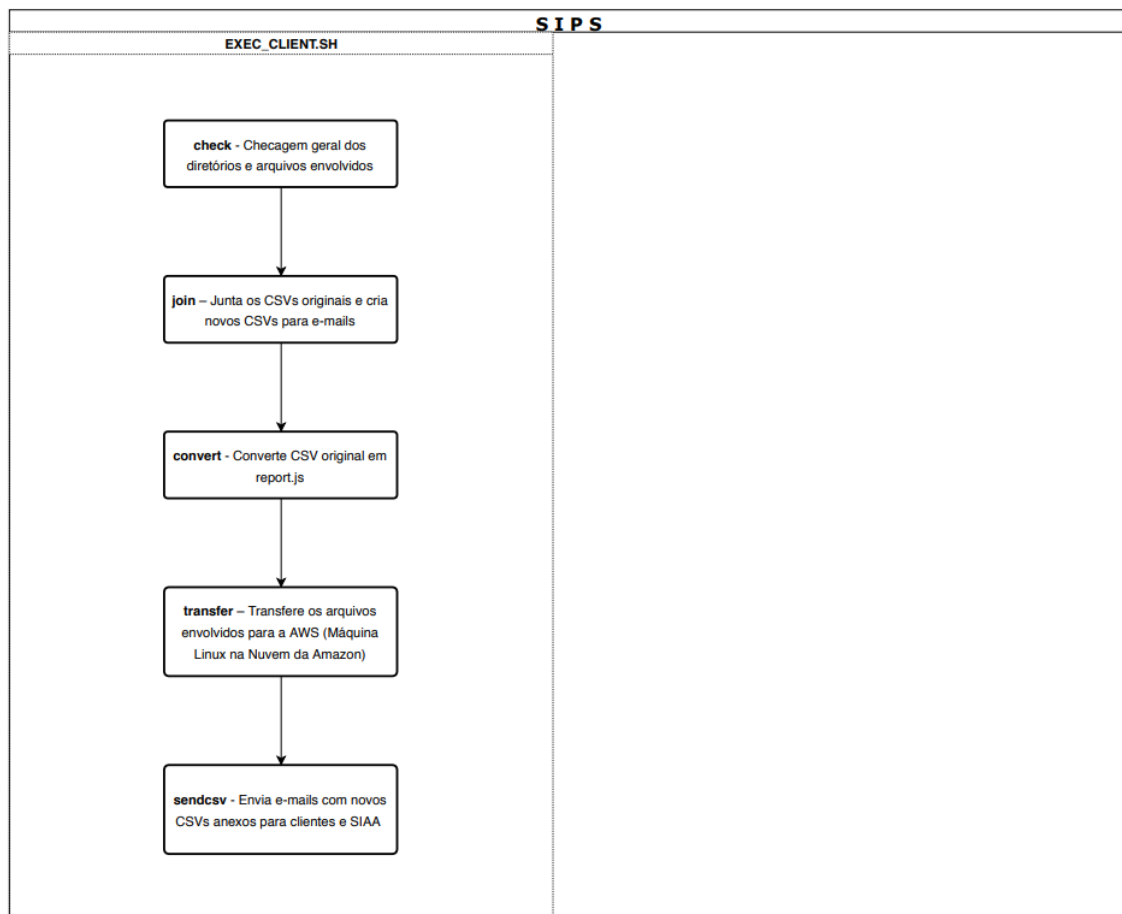


Diagrama 4 – Detalhe três – Exec_client.sh - Nautilus



Implementação de novo cliente

O projeto SIPS foi praticamente todo desenvolvido em ambientes LINUX, com o intuito de ser o mais automático possível e com a menor intervenção humana possível, no momento atual do projeto.

Três clientes fictícios foram criados nos sistemas LINUX do projeto para se obter um bom e consistente MVP (Minimum Viable Product – Mínimo Produto Viável). Os nomes desses clientes são 'SS', 'DEMO' e 'PUBLIC'.

Todos os processamentos envolvidos foram testados exaustivamente com os três clientes acima com sucesso e estão estabelecidos e rodando automaticamente todos os dias.

Para a implementação de um cliente NOVO, de nome fictício 'XXXXX' a ser trocado obviamente pelo nome real do novo cliente, será preciso a execução manual dos passos listados abaixo. O tempo estimado de execução de todos os passos é de 30-60 minutos.

São passos muito simples e usuais aos ambientes LINUX, como criar diretórios, copiar arquivos, duplicar e alterar linhas de rotinas e arquivos. Vale notar que, em uma eventual fase futura desse projeto, todos esses passos poderiam ser também eles automatizados.

Ao final da execução com sucesso de TODOS os passos abaixo listados, o cliente NOVO 'XXXXX' estará totalmente implementado e será processado automaticamente na próxima rodada diária de processamentos, juntamente com os outros clientes já existentes.

- No servidor **NAUTILUS**.

1. Criar e incluir rotina octave para o novo cliente no script **exec_meteo.sh** conforme exemplo abaixo.

```
.1.1. # WRF Pos Proc 004XXXXX (retirado de grade Atlas)
      octave --no-gui --silent $FOLATLAS/automata/posproc_wrf_json2_004XXXXX.m >> $LOGD 2>&1
```

2. Criar e incluir rotina octave para o novo cliente no script **exec_swan.sh** conforme exemplo abaixo.

```
.2.1. # WW3 Pos Proc 004XXXXX
      octave --no-gui --silent $folatlas/automata/posproc_ww3_json2_004XXXXX.m >> $LOGD 2>&1
```

3. Incluir linha da função 'sendcsv' para o novo cliente no final do script **exec_client.sh** conforme exemplo abaixo.

```
.3.1. # EMAIL_CLIENT $1 EMAIL_SIAA $2 SUBJECT $3 DIR-CLI $4 BODY OF
      EMAIL... $5
      sendcsv "operational@siaabr.com" "operational@siaabr.com" "Report do dia $(date +%d/%m/%y) - Porto do
      Cliente XXXXX" "004XXXXX" "Reports do dia $(date +%d/%m/%y) em anexo.^M^MVisualize também no sistema
      web: http://XXXXX.modsiaabr.com/tabler.html"
```

4. Criar diretório do novo cliente no /home/nemo/rooms/steaks/ conforme exemplo abaixo.

```
.4.1. /home/nemo/rooms/steaks/
      drwxr-xr-x 8 nemo nemo 4096 Ago 23 11:39 001public
      drwxr-xr-x 8 nemo nemo 4096 Jun 25 20:34 002ss
      drwxrwxr-x 8 nemo nemo 4096 Jun 26 10:49 003demo
      drwxrwxr-x 2 nemo nemo 4096 Jun 26 10:47 004XXXXX
```

5. Criar subdiretórios do novo cliente no /home/nemo/rooms/steaks/004XXXXX conforme exemplo abaixo.

```
.5.1. drwxr-xr-x 2 nemo nemo 4096 Ago 24 18:17 webgrids_swan
      drwxr-xr-x 2 nemo nemo 4096 Ago 24 14:56 webgrids_wrf
      drwxr-xr-x 2 nemo nemo 4096 Ago 27 18:49 webpoints_swan
      drwxr-xr-x 2 nemo nemo 4096 Ago 27 18:50 webpoints_wrf
      drwxr-xr-x 2 nemo nemo 4096 Ago 23 11:39 webrisks
      drwxr-xr-x 2 nemo nemo 4096 Set 5 11:09 webtabs
```

- Na **AWS** (Máquina Virtual Linux na nuvem da AWS)

1. Acessar a instância Linux da SIAA na AWS usando o procedimento abaixo.

```
=====
=====
```

Para acessar arquivos dos sites servidos pela Amazon, siga os seguintes passos:

1. Instale a chave de acesso no computador que fará login na instância EC2:

1.1 Faça o download do arquivo **msaws.pem** da chave de acesso, e mova-a para o diretório `.ssh`

```
$> mv msaws.pem ~/.ssh
```

1.2 Mude as permissões do arquivo:

```
$> chmod 0400 ~/.ssh/msaws.pem
```

1.3 **Este procedimento é necessário apenas uma vez, porém em todo computador que for acessar a instância.**

2. Acesse a instância:

```
$> ssh -i ~/.ssh/msaws.pem ec2-user@modsiaabr.com
```

3. Os arquivos servidos no site estão em:

```
/var/www/html/
```

```
=====
=====
```

2. Criar diretório do novo cliente `/var/www/html/XXXXXX` conforme exemplo abaixo.

2.1. `/var/www/html/`

```
drwxrwxr-x 8 apache apache 4096 Feb 26 2018 siaa
drwxrwxr-x 11 apache apache 4096 Mar 13 13:15 sepetiba
drwxrwxr-x 7 ec2-user ec2-user 4096 Apr 9 13:53 sprj1_OLD
drwxrwxr-x 7 ec2-user ec2-user 4096 Apr 10 15:35 portoacu_OLD
drwxrwxr-x 7 apache apache 4096 Aug 10 15:07 portoacu
drwxrwxr-x 11 apache apache 4096 Aug 10 18:30 ss
drwxrwxr-x 11 apache apache 4096 Aug 13 14:18 demo
drwxrwxr-x 11 apache apache 4096 Aug 24 12:50 public
drwxrwxr-x 11 apache apache 4096 Aug 24 12:50 XXXXX
```

3. Copiar todos os arquivos e todos os subdiretórios de um cliente anterior para o diretório do novo cliente `/var/www/html/XXXXXX` conforme exemplo abaixo.

3.1. `/var/www/html/public`

```
[ec2-user@ip-172-31-17-88 public]$ ls -ltr
-rw-rw-r-- 1 ec2-user ec2-user 20707 Feb 26 2018 tabler.html
-rw-rw-r-- 1 ec2-user ec2-user 468 Feb 26 2018 statgraph.html
-rw-rw-r-- 1 ec2-user ec2-user 384 Feb 26 2018 reliable.html
-rw-rw-r-- 1 ec2-user ec2-user 666 Feb 26 2018 dynmaps.html
-rw-rw-r-- 1 ec2-user ec2-user 24879 Feb 26 2018 combograph.html
-rw-rw-r-- 1 ec2-user ec2-user 906 Feb 26 2018 radar.html
drwxrwxr-x 2 apache apache 4096 Mar 13 13:27 js
drwxrwxr-x 2 apache apache 4096 Apr 13 14:41 css
drwxrwxr-x 2 ec2-user ec2-user 4096 Apr 13 14:41 coda
drwxrwxr-x 10 ec2-user ec2-user 4096 Apr 13 14:41 glyphs
-rw-rw-r-- 1 ec2-user ec2-user 3923 Apr 17 17:08 index.html
-rw-rw-r-- 1 ec2-user ec2-user 105 Apr 17 17:11 risc.html
drwxrwxr-x 7 apache apache 4096 Apr 17 17:50 src
drwxrwxr-x 3 ec2-user ec2-user 4096 Jul 16 22:36 code
-rw-rw-r-- 1 ec2-user ec2-user 17280 Jul 29 23:34 maps.html
drwxrwxr-x 2 apache apache 4096 Aug 13 19:49 img
drwxrwxr-x 4 apache apache 4096 Aug 13 20:10 data
drwxrwxr-x 2 ec2-user ec2-user 4096 Aug 13 20:11 files
```

4. Criar 'subdomínio' do novo cliente **XXXXXX** na AWS usando o procedimento abaixo.

=====

- 1- Acessar a instância Linux AWS usando o procedimento acima [no item 1](#).
- 2- Ir para o diretório `/etc/httpd/conf.d/`
- 3- Digitar o comando `sudo su` para obter permissões
- 3- Copiar por EXEMPLO o arquivo `04_demo.conf` para novo arquivo `05_XXXXX.conf`
- 4- Editar o novo arquivo e alterar toda ocorrência de `demo` para `XXXXX`
- 5- Salvar e sair do arquivo.
- 6- Acessar <https://aws.amazon.com/pt/console/> (Usuário e Senha do Felipe)
- 7- Ir em *Services - Route 53 - Hosted Zones - modsiaabr.com - Create Record Set*
- 8- No campo *NAME* digitar o novo nome, por EXEMPLO **XXXXXX**.

No campo *VALUE* digitar o endereço de IP da instância Linux na AWS (18.231.153.124)

- 9- Clicar no botão *CREATE*

10-Depois de alguns minutos testar no navegador o novo endereço, por EXEMPLO, **XXXXXX.modsiaabr.com**

=====

5. Colocar senha de acesso ao site do novo cliente **XXXXXX** na AWS usando o procedimento abaixo.

=====

- 1- Acessar a instância Linux AWS usando o procedimento acima [no item 1](#).
- 2- Ir para o diretório `/var/www/admin/`
- 3- Digitar o comando `sudo su` para obter permissões
- 4- Digitar o comando `cat htpasswd` para ver os usuários com senha até o momento
- 5- Digitar o comando `htpasswd /var/www/admin/htpasswd XXXXXX`
- 6- Digitar a senha e confirmar a senha.

=====

4. Conclusão do MVP

Dentro do contexto descrito em todos os itens anteriores deste relatório podemos concluir com satisfação que o objetivo inicial de obtenção de um MVP consistente para o SIPS, e os respectivos desafios associados a ele, foi totalmente alcançado.

Não apenas um mas TRÊS clientes fictícios foram criados nos dois ambientes do projeto, no sistema LINUX e no sistema WEB, sistemas esses que estão perfeitamente integrados.

Todos os processamentos envolvidos foram testados exaustivamente para os três clientes com sucesso e estão estabelecidos e rodando automaticamente todos os dias.

5. Ganhos na formação profissional

Durante o excelente período de trabalho, pesquisa e desenvolvimento que esta bolsa me proporcionou não apenas obtive muitos ganhos à minha formação profissional como também à minha formação pessoal.

Os enormes desafios propostos inicialmente foram mais do que suficientes para que, com esforços igualmente enormes, eu pudesse evoluir muitíssimo tecnicamente e pessoalmente.

6. Considerações finais e perspectivas futuras

Após trabalhar por mais de 20 anos na indústria e/ou no mercado de trabalho tradicional, posso dizer que foi a realização de um sonho trabalhar com Pesquisa.

Gostaria de registrar aqui minha imensa gratidão a SIAA e a FAPESP.

7. Referências bibliográficas

NEVES, Julio Cezar. **Programação SHELL LINUX**. 10. ed. Rio de Janeiro: BRASPORT, 2015. 511 p. v. 1.

ANEXO 1

Scripts inteiros do sistema em funcionamento em Agosto de 2017.

EXECUTOR.PL

```
=====

#!/usr/bin/perl

use strict;
use IO::File;

require "time.pllib";

my $false = undef;
my $true  = 1;
my $logplace = "/rooms/engine/logs/";
my $logname = "executor.log";
my $CLIENTSHOME = $ENV{LOCALFARM};
my $BOOTLAIR = "/rooms/engine/automata/";
my $LOGHANDLE;

main( $ARGV[0], $ENV{PWD} );
exit 0;

sub main
{
    my $hl;
    my $idx;
    my @list;
    my @fpbots;
    my @bots;
    my $item;
    my $ftmp = `tempfile`;
    my $logfile;
    my $client;
    my ( $kind, $place ) = @_ ;

    $place = $BOOTLAIR if not defined $place;
    open $hl, ">>${place}/../logs/${logname}";
    $LOGHANDLE = $hl;
    dolog( "main received: arg1: '$kind' and arg2: '$place'." );
    locatefiles( $place, \@list, \@fpbots, $kind );

    chdir $place;
    dolog( "the following bot(s) was(ere) found: " );
    @bots = sort @list;
    for $item( @bots ) { dolog( "\t >> $item" ); }
    dolog( "they will be executed in that order momentarily." ) if $#list >= 0;
    for $idx( 0 .. $#bots )
    {
        $item = $bots[ $idx ];
        dolog( "about to call $item" );
        if( $item =~ /do_client_(\w+)/ )
        {
            $client = $1;
            $ENV{LOCALEGG}=$client;
            if( !check_environment() ) { dolog( "local environment not set!" ); }
            else
            {
                dolog( "client '$client' particular robots detected." );
                dolog( "changing execution place to: '${CLIENTSHOME}$client'" );
                dolog( "informing lorpa the toca is located now at:
${CLIENTSHOME}$client" );
                main( undef, "${CLIENTSHOME}$client/automata" );
            }
        }
        system $fpbots[$idx] ." &> $ftmp";
        dolog( "$item reported the following in its execution:" );
        dolog( "\t>>", $ftmp );
        dolog( "end report for $item" );
        unlink $ftmp;
    }
}
```

```

        sleep 5;
    }
    dolog( "executor ends." );
    close $hl;
    return $true;
}

sub check_environment
{
    my $src = 0;

    if( $ENV{LOCALETC} eq undef ) { $src += 0x01; dolog("localetc not set!"); }
    if( $ENV{LOCALLIB} eq undef ) { $src += 0x02; dolog("locallib not set!"); }
    if( $ENV{LOCALBIN} eq undef ) { $src += 0x04; dolog("localbin not set!"); }
    if( $ENV{LOCALFARM} eq undef ) { $src += 0x08; dolog("localfarm not set!"); }

    if( $src == 0 ) { return $true; } else { return $false; }
}

sub locatefiles
{
    my @Folders;
    my @Files;
    my $file;
    my $ffile;
    my $item;
    my $kind;
    my ( $folder, $list, $names, $criteria ) = @_;

    dolog( "entering folder: '$folder' with criteria: '$criteria'" );
    dirinfo( \@Folders, \@Files, $folder, $false );
    $kind = substr $criteria, 0, 1 if $criteria ne undef;

    for $ffile ( @Files )
    {
        $file = $ffile;
        $file =~ s/$folder[/\]*///;
        next if $file !~ /^[hwm]*\d\d\_/;
        next unless -x $ffile;
        if( defined $criteria ) { push @$list, $file if $file =~ /^$kind/; }
        else
        {
            next if $file =~ /\^D/;
            push @$list, $file;
            push @$names, "$folder/$file";
        }
    }
    dolog( "leaving folder '$folder'." );
}

sub dirinfo
{
    my $F;
    my $hDIR;
    my @Files;
    my $subdircnt;
    my( $Dirs, $Files, $Dir, $recurse, $cntdir ) = @_;

    if( not defined $cntdir ) { $subdircnt = 1; } else { $subdircnt = $cntdir; }
    opendir( $hDIR, $Dir ) or warn "Could not open $Dir ($!)\n";
    @Files = readdir( $hDIR );
    closedir( $hDIR );

    foreach $F ( @Files )
    {
        next if( $F eq "." );
        next if( $F eq ".." );
        $F = "$Dir/$F";
        if( ! -d $F ) { push @Files, $F; }
        else
        {
            $subdircnt++;
            if( $recurse )
            {
                #
                last if $subdircnt >= $MAX_SUBDIR_STRUCTURE;
                dirinfo( $Dirs, $Files, $F, $recurse, $subdircnt );
            }
        }
    }
}

```

```

    }
    push @$Dirs, $F;
}

}

sub dolog
{
    my $line;
    my $hi;
    my $diem = datefmt( "d" );
    my $hora = datefmt( "t" );
    my ( $msg, $flag ) = @_ ;

    if( not defined $flag ) { printf $LOGHANDLE "%8s\t%8s\t%s\n", $diem, $hora, $msg; }
    else
    {
        open $hi, $flag;
        while( $line=<$hi> ) { print $LOGHANDLE "$msg $line"; }
        close $hi;
    }
}
}
=====

```

DESGRIBIT

```

=====

#!/bin/bash
#####
#
# this software is released under the GPL3 terms
# (L)copyleft 2015 RobertoL Parra - RLPARRA@GMAIL.COM
#
# developed by RobertoL Parra for SIAA under terms of GPL3
#
#
# --
# desgribit is an utility to construct a data base from grib or grib2 data files
# it extracts the vars available and expells a line with them. This version
# provisions OPT variables to be received as command line arguments, but not yet
# implemented. So, latitude and longitude values are for now, hardcoded.
# --
#####
LOGFILE="/rooms/engine/logs/desgribit.log";
BOTLAIR="/rooms/engine/automata/";
REPO="/rooms/engine/wavewatch/";
UNGRIB="${BOTLAIR}/wgrib2";
ORDER="WIND WDIR UGRD VGRD HTSGW PERPW DIRPW VVHGT SWELL1 SWELL2 WVPER SWPER1 SWPER2 WVDIR SWDIR1
SWDIR2"

RUN="";
LINE="";
WWFILE="";
WWBASE="";
HEADER="";
FILE="";
declare -a VALUES=();

OPT_RUN=""
OPT_DATA=""
OPT_LATITUDE=""
OPT_LONGITUDE=""
OPT_START=0;
OPT_END=180;
OPT_STEP=3;
OPT_SEPARATOR=":";
OPT_OUTPUT="";
OPT_ORDER="";

dolog()
{
    local MSG="$@";
    local DIEM=`date +%Y%m%d\t%H%M%S`;
}

```

```

    echo -e "$DIEM\t$MSG" >> $LOGFILE;
}

wwfilefinder()
{
    local FOUND=0;
    local NUM;
    local FILE;
    local RDIEM;
    local R;

    for R in 18 12 6 0; do
        NUM=`printf "%02d" $R`;
        RDIEM="wavewatch-${DIES}";
        FILE="{REPO}/{RDIEM}/{RDIEM}-${NUM}-000.grib2";
        dolog "testing wwfilefinder for '$FILE'";
        test -f "$FILE" || continue;
        dolog "found '$FILE' in wwfilefinder";
        WWFILE="$FILE";
        WWBASE="{REPO}/{RDIEM}/{RDIEM}-${NUM}";
        RUN=$R;
        FOUND=1;
        break;
    done
    test "$FOUND" = "0" && { dolog "not found index file."; exit 1; }
}

fieldsfinder()
{
    local I;
    local H="";
    local ITEM;
    local FIELDS;
    local FILE="$WWFILE";

    FIELDS=( `$UNGRI $FILE | cut -d ":" -f 4` );
    for (( I=0; I<${#FIELDS[*]}; I++ )); do
        if test "${FIELDS[I]}" != "${FIELDS[I+1]}"; then
            ITEM="${FIELDS[I]}";
        else
            ITEM="${FIELDS[I]}1${SEPARATOR}${FIELDS[I+1]}2";
            let I++;
        fi;
        test -z "$ITEM" && dolog "error parsing header: empty field at I=$I";
        test -z "$H" && H="$ITEM" || H="{H}${SEPARATOR}${ITEM}";
    done
    test -z "$H" && { dolog "fieldsfinder not found header."; exit 1; }
    REPLY="$H";
    HEADER="$H";
}

fieldsorder()
{
    local FIELD="$1";
    local ITEM;
    local COUNT=0;
    local FOUND=0;
    local ifs;

    ifs=$IFS;
    IFS=':';

    for ITEM in $HEADER; do
        test "$FIELD" = "$ITEM" && { FOUND=1; break; };
        let COUNT++;
    done
    IFS=$ifs;
    test "$FOUND" = "1" && return $COUNT;
    dolog "field '$FIELD' not found in header '$HEADER'";
}

valuesfinder()
{
    local LINE;
    local IDX=0;

```



```

local FILE="$1";
local LINES=();

if test -f "${FILE}"; then
    LINES=(`SUNGRIB -lon ${LATITUDE} ${LONGITUDE} ${FILE}`);
else
    for LINE in $ORDER; do LINES[IDX]="_"; let IDX++; done
fi

for LINE in ${LINES[@]}; do
    VALUES[IDX]="${LINE/*val=}";
    let IDX++;
done

}

datafinder()
{
    local N;
    local LINE;
    local ITEM;
    local FILE="$1";

    valuesfinder "$FILE";
    for ITEM in $ORDER; do
        fieldsorder $ITEM
        N="$?";
        test -z "$LINE" || LINE="${LINE}${SEPARATOR}${VALUES[N]}";
        test -z "$LINE" && LINE="${VALUES[N]}";
    done

    REPLY="${LINE//_/_}";
}

expell()
{
    local FILE="$1";

    if test "$OUTPUT" = "STDOUT"; then
        cat $FILE;
    else
        if test -z "$OPT_ORDER"; then
            command mv "$FILE" "${WWBASE}-run.dat";
        else
            if test "${OPT_ORDER:0:1}" == "/"; then
                command mv "$FILE" "${OPT_ORDER}.dat";
            else
                command mv "$FILE" "${WWBASE}-${OPT_ORDER}.dat";
            fi
        fi
    fi;
}

datamaker()
{
    local I;
    local FILE;
    local S="${SEPARATOR}";
    local OUTP=`tempfile`;

    for (( I=$OPT_START; I<$OPT_END; I+= $OPT_STEP )); do
        IDX=`printf "%03d" "$I"`;
        FILE="${WWBASE}-${IDX}.grib2"
        datafinder "$FILE";
        echo "${DIES}${S}${RUN}${S}${IDX}${S}$REPLY" >> $OUTP;
    done
    REPLY="$OUTP";
}

init()
{
    while test -n "$1"; do
        case "$1" in
            -r|--run)
                shift;
                OPT_RUN="$1";
            ;;

```

```

-d|--date)
    shift;
    OPT_DATA="$1";
;;

-l|--latitude)
    shift;
    OPT_LATITUDE="$1";
;;

-L|--longitude)
    shift;
    OPT_LONGITUDE="$1";
;;

-s|--separator)
    shift;
    OPT_SEPARATOR="$1";
;;

-o|--output)
    OPT_OUTPUT="STDOUT";
;;

-O|--order)
    shift;
    OPT_ORDER="$1";
;;

-h|--help)
    echo "Syntax: ${0##/*} [options]";
    echo "where options can be:";
    echo "  r run      arg  arg is the run ";
    echo "  d date     arg  arg is the date";
    echo "  l --latitude arg  arg is the latitude";
    echo "  L --longitude arg  arg is the lngitude";
    echo "  s --separator arg  arg is the field separator";
    echo "  o --output   sets the output to be STDOUT";
    echo "  O --order    arg  [/path]arg = output filename";
    echo "  h help      shows this text";
;;

*) echo "option not recognized!";
;;

esac;
shift;
done

test "$OPT_RUN" = "" && RUN="00" || RUN=`printf "%02d" "$OPT_RUN"`;
test "$OPT_DATA" = "" && DIES=`date +%Y%m%d` || DIES="$OPT_DATA";
test "$OPT_LATITUDE" = "" && LATITUDE="-24" || LATITUDE="$OPT_LATITUDE";
test "$OPT_LONGITUDE" = "" && LONGITUDE="-43.5" || LONGITUDE="$OPT_LONGITUDE";
test "$OPT_SEPARATOR" = "" && SEPARATOR=":" || SEPARATOR="$OPT_SEPARATOR";
}

init "$@";
wwfilefinder;
fieldsfinder;
datamaker;
expell "$REPLY";
test -f "$REPLY" && command rm "$REPLY";

=====

```

20_PRUMO_CONVERT

```

=====

#!/bin/bash
#####
# DO_CONVERT picks up the vars file from desgribit and do a little processing
# inside them, using octave.
#
#
# Author: RobertoL Parra

```

```

# Copyright (C) 2001-2015 RobertoL Parra <rlparra@gmail.com>
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
#
# (C)Copyright 2015 SIAA - RobertoL Parra - rlparra@gmail.com
#####
REPO="/rooms/engine/wavewatch/";
DREPO="/rooms/engine/swash/";
BOTLAIR="/rooms/engine/automata/";
OUTSWASHREPO="${DREPO}outputs/";
MSCRIPT="faz_cc.m";
TMPFILE=`tempfile`;
DIEM=$(date +%Y%m%d);
HORA=$(date +%H%M%S);
ROTA="${REPO}wavewatch-${DIEM}";
OFILE="";
DFILE="";

if test "${PWD}/rooms/suits" == "${PWD}"; then
    CLIENT="TMP=( echo ${0//_/ } ); echo ${TMP[2]}";
else
    CLIENT="$N="${PWD}/rooms/automata"; echo ${N##*/}";
fi

BOTLAIR="/rooms/engine/automata/";
CLIENTHOME="/rooms/suits/${CLIENT}/data/"
LOGFILE="/rooms/suits/${CLIENT}/logs/doconvert.log";

dolog()
{
    local MSG="$@";

    echo -e "${DIEM}\t${HORA}\t${MSG}" >> $LOGFILE;
}

cd "${DREPO}";
test -d "wavewatch-${DIEM}" || mkdir "wavewatch-${DIEM}";

dolog "doconvert called at $DIEM : $HORA";

if test ! -d "${ROTA}"; then
    dolog "Run folder not found.";
    dolog "folder tryed: '$ROTA'";
    dolog "convert.m was not called.";
else
    cd ${ROTA};
    dolog "changed to '$ROTA'";
    echo "TPAR" > $TMPFILE;
    octave "${BOTLAIR}${MSCRIPT}" "${CLIENT}" "${CLIENTHOME}" &>> $TMPFILE;
    dolog "$MSCRIPT called with arg0: '$CLIENT' && arg1: '$CLIENTHOME'";
    dolog "--- octave begin ---"
    dolog "`cat $TMPFILE`";
    dolog "--- octave end ---";

    OFILE="${DREPO}latest_run.csv"
    DFILE="${OUTSWASHREPO}${DIEM}-${HORA}.csv";

    if test -f "$OFILE"; then
        cp "$OFILE" "$DFILE";
        chgrp "siaa" "$DFILE";
        chmod g+w "$DFILE";
        dolog "convert successfully run: generated $DFILE";
    fi
fi
#####

```

ANEXO 2

Arquivos HTML inteiros do novo sistema WEB.

MAPS.HTML

```
=====

<!DOCTYPE html>

<!-- ===== SIAABR.COM - Alex Luz - 24/07/2018 ===== -->

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>DEMO HOME SIPS</title>
    <link rel="stylesheet" href="src/leaflet.css">
    <link rel="stylesheet" href="src/css/bootstrap.css">
    <link rel="stylesheet" href="src/plugins/L.Control.Zoomslider.css">
    <link rel="stylesheet" href="src/plugins/L.Control.MousePosition.css">
    <link rel="stylesheet" href="src/plugins/L.Control.MapCenterCoord.css">
    <link rel="stylesheet" href="src/plugins/leaflet-sidebar.css">
    <link rel="stylesheet" href="src/plugins/easy-button.css">
    <link rel="stylesheet" href="src/plugins/leaflet-velocity.css">
    <link rel="stylesheet" href="src/plugins/leaflet.timedimension.control.css">

    <link rel="stylesheet" href="https://api.mapbox.com/mapbox.js/plugins/leaflet-
fullscreen/v1.0.1/leaflet.fullscreen.css"/>
    <link rel="stylesheet" href="http://ljangis.github.io/leaflet-measure/leaflet-measure.css" />

    <script src="src/leaflet-src.js"></script>
    <script src="src/jquery-3.2.0.min.js"></script>
    <script src="src/plugins/L.Control.Zoomslider.js"></script>
    <script src="src/plugins/L.Control.MousePosition.js"></script>
    <script src="src/plugins/L.Control.MapCenterCoord.js"></script>
    <script src="src/plugins/leaflet-sidebar.js"></script>
    <script src="src/plugins/easy-button.js"></script>
    <script src="src/plugins/leaflet-providers.js"></script>

    <script src="https://api.mapbox.com/mapbox.js/plugins/leaflet-
fullscreen/v1.0.1/Leaflet.fullscreen.min.js"></script>
    <script src="http://ljangis.github.io/leaflet-measure/leaflet-measure.js"></script>

    <script src="src/plugins/LeafletPlayback.min.js"></script>

<!-- ***** ALTERA PALAVRA 'Speed' PARA 'Height' SOMENTE PARA ONDAS-WAVES - ALEX LUZ - 06/04/2018. --->
<!-- ***** ALTERAÇÃO ESTÁ DENTRO DO CÓDIGO ABAIXO DA VELOCITY - "src/plugins/leaflet-velocity.js". --->
    <script src="src/plugins/leaflet-velocity.js"></script>

    <script type="text/javascript" src="https://cdn.rawgit.com/nezasa/iso8601-js-
period/master/iso8601.min.js"></script>
    <script src="src/plugins/leaflet.timedimension.js"></script>
    <script src="src/plugins/leaflet.timedimension.util.js"></script>
    <script src="src/plugins/leaflet.timedimension.layer.js"></script>
    <script src="src/plugins/leaflet.timedimension.layer.wms.js"></script>
    <script src="src/plugins/leaflet.timedimension.layer.geojson.js"></script>
    <script src="src/plugins/leaflet.timedimension.player.js"></script>
    <script src="src/plugins/leaflet.timedimension.control.js"></script>

    <style>
      #mapdiv {
        height:100vh;
      }
    </style>
  </head>
  <body>
    <div id="mapdiv" class="col-md-12"></div>
    <script>
      var mymap;
      var lyrOSM;
      var lyrOSMBW;
      var lyrOTM;
      var lyrOCM;
      var lyrTrans;
      var lyrOutdoors;
      var lyrPioneer;
      var lyrLandscape;
      var lyrESRIWMS;
      var lyrESRINatGeo;
      var lyrESRITopo;
      var lyrESRIImagery;
      var lyrESRIGrey;
      var lyrESRIOcean;
      var lyrESRIDelorme;
      var lyrWatercolor;
      var lyrToner;
      var lyrTerrain;
      var lyrHereBasic;
      var lyrHereHybrid;
      var lyrHereTerrain;
      var lyrVentoReal;
      var lyrOndaReal;
      var mrkCurrentLocation;
```



```

        ctlEasyButtonOR = L.easyButton({
        states: [{
            stateName: 'Add-OndaReal',
            icon: 'glyphicon-globe',
            title: 'VER - Oceano - Ondas/Correntes',
            onClick: function() {
                arrvelwave[i].addTo(mymap);
                ctlEasyButtonOR.state('Rem-OndaReal');
                STATUS = "WAVE";
                mymap.addControl(timeDimensionControlWW);
                ctlEasyButtonVR.disable();
            }
        }, {
            stateName: 'Rem-OndaReal',
            icon: 'glyphicon-refresh',
            icon: 'glyphicon-globe',
            title: 'RETIRAR - Oceano - Ondas/Correntes',
            onClick: function() {
                mymap.removeLayer(arrvelwave[i]);
                ctlEasyButtonOR.state('Add-OndaReal');
                STATUS = "";
                mymap.removeControl(timeDimensionControlWW);
                ctlEasyButtonVR.enable();
            }
        }
    ]});

    ctlEasyButtonOR.addTo(mymap);

// ***** BOTAO WIND *****

    ctlEasyButtonVR = L.easyButton({
    states: [{
        stateName: 'Add-VentoReal',
        icon: 'glyphicon-cloud',
        title: 'VER - Atmosfera - Ventos Médios/Rajadas',
        onClick: function() {
            arrvelwind[i].addTo(mymap);
            ctlEasyButtonVR.state('Rem-VentoReal');
            STATUS = "WIND";
            mymap.addControl(timeDimensionControlWW);
            ctlEasyButtonOR.disable();
        }
    }, {
        stateName: 'Rem-VentoReal',
        icon: 'glyphicon-refresh',
        icon: 'glyphicon-cloud',
        title: 'RETIRAR - Atmosfera - Ventos Médios/Rajadas',
        onClick: function() {
            mymap.removeLayer(arrvelwind[i]);
            ctlEasyButtonVR.state('Add-VentoReal');
            STATUS = "";
            mymap.removeControl(timeDimensionControlWW);
            ctlEasyButtonOR.enable();
        }
    }
    ]});

    ctlEasyButtonVR.addTo(mymap);

    ctlAttribute = L.control.attribution({position:'bottomright'}).addTo(mymap);
    ctlAttribute.addAttribution('&copy; <a href="http://siaabr.com">SIAA</a>');

    ctlScale = L.control.scale({position:'bottomleft'}).addTo(mymap);

    mymap.on('contextmenu', function(e) {
        var dtCurrentTime = new Date();

L.marker(e.latlng).addTo(mymap).bindPopup(mymap.getBounds().toString()+"<br>"+e.latlng.toString()+"<br>"+dtCurrentTime.toString());

    });

// ***** VELOCITY *****

// load data (u, v grids) from somewhere (e.g. https://github.com/danwild/wind-js-server) - ***** W I N D
*****
for (i=1;i<42;i++)
    (function(i) {
        $.getJSON('./data/wind/latest_meteo'+[i]+'json', function (data) {

            arrvelwind[i] = L.velocityLayer({
                displayValues: true,
                displayOptions: {
                    velocityType: 'GBR Wind',
                    displayPosition: 'bottomleft',
                    angleConvention: 'meteoCW',
                    displayEmptyString: 'No wind data',
                    speedUnit: 'kt'
                },
                data: data,
                maxVelocity: 25,
                minVelocity: 0,
                velocityScale: 0.002 // arbitrary default 0.005
            });
        });
    })(i);

// load data (u, v grids) from somewhere (e.g. https://github.com/danwild/wind-js-server) - ***** W A V E
*****
for (i=1;i<42;i++)

```

```

(function(i) {
    $.getJSON('./data/wave/latest_wave'+[i]+'json', function (data) {

        arrvelwave[i] = L.velocityLayer({
            displayValues: true,
            displayOptions: {
                velocityType: 'GBR Wave',
                displayPosition: 'bottomleft',
                angleConvention: 'meteoCW',
                displayEmptyString: 'No wave data',
                speedUnit: 'm'
            },
            data: data,
            maxVelocity: 5,
            minVelocity: 0,
            velocityScale: 0.05 // arbitrary default 0.005
        });
    });
})(i);

// ***** TIME DIMENSION *****

var CurrentDate = new Date();
var AAAA = CurrentDate.getFullYear();

var DDMMHH = Data[22]; // ##### VAR 'Data'(DADOS) VEM DO 'REPORT.JS' - PRIMEIRO 'DDMMHH' DO REPORT.JS
// ESTÁ NA POSIÇÃO 22 DO ARRAY 'DATA' #####
var DD = DDMMHH.substr(0,2);
var MM = DDMMHH.substr(2,2);
var HH = DDMMHH.substr(4,2);

var startDate = new Date(AAAA, MM-1, DD, HH, 00); // ##### Data inicial = DDMMHH do report.js conforme
// comentario acima #####

// start of TimeDimension manual instantiation - ***** WW *****

timeDimensionWW = new L.TimeDimension({
    timeInterval: startDate.toISOString() + "/PT120H",
    period: "PT3H"
});

playerWW = new L.TimeDimension.Player({
    transitionTime: 1000,
    loop: false,
    startOver: true
}, timeDimensionWW);

timeDimensionControlOptionsWW = {
    playButton: false,
    player: playerWW,
    timeDimension: timeDimensionWW,
    position: 'bottomleft',
    autoPlay: false,
    timeSliderDragUpdate: false,
    speedSlider: false,
    timeZones: ["Local", "UTC"]
// timeZones: ["Local"]
};

timeDimensionControlWW = new L.Control.TimeDimension(timeDimensionControlOptionsWW);

i=1;
timeDimensionWW.on('timeload', function () {

    if (STATUS == "WAVE") {
        if (arrvelwave[i]) {
            mymap.removeLayer(arrvelwave[i]);
        };
        arrvelwave[i+1].addTo(mymap);
        i++;
    }
    if (STATUS == "WIND") {
        if (arrvelwind[i]) {
            mymap.removeLayer(arrvelwind[i]);
        };
        arrvelwind[i+1].addTo(mymap);
        i++;
    }
});

}); // ***** END OF $(document).ready(function(){ *****

</script>

<script src='code/report.js'></script> <!-- //##### 'REPORT.JS' PARA PEGAR PRIMEIRA
'DDMMHH' ##### -->

</body>
</html>

```

REPORT.JS

```
=====

var Data = [
"data","Hsto","Tpto","Dmto","Vv","Dv","Raj","Temp","Prec","Nuv","Hum",
"ddmmHH","m","s","°","nos","°","nos","°C","mm","%","%",
"240703","0.02","2.1","0.0","5.7","51.9","7.0","19.2","0.0","0.0","82.2",
"240706","0.01","1.9","316.3","1.7","319.8","2.1","18.2","0.0","7.9","85.2",
"240709","0.01","1.9","288.5","10.1","297.1","12.5","18.9","0.1","77.8","95.4",
"240712","0.01","1.9","281.7","7.7","275.3","9.6","19.7","0.0","78.6","92.5",
"240715","0.00","1.9","278.8","5.1","247.2","6.3","20.0","0.0","74.2","91.0",
"240718","0.00","1.5","277.3","0.9","260.8","1.2","18.8","0.0","83.7","93.9",
"240721","0.00","1.1","275.1","0.5","301.1","0.6","18.4","0.0","77.4","91.0",
"250700","0.00","5.7","246.7","0.7","46.0","0.9","18.4","0.0","78.6","92.5",
"250703","0.03","5.7","219.4","2.9","77.0","3.6","18.4","0.0","83.7","92.5",
"250706","0.16","5.1","224.5","3.0","79.0","3.7","17.9","0.0","73.0","88.5",
"250709","0.35","4.5","221.9","2.4","163.4","3.1","19.6","0.0","89.7","90.3",
"250712","0.38","4.5","216.6","4.6","173.5","5.7","19.7","1.2","96.0","95.5",
"250715","0.42","4.5","202.5","3.6","156.0","4.4","20.1","0.2","90.9","93.9",
"250718","0.52","5.1","186.3","2.0","112.6","2.5","19.0","0.0","73.0","94.5",
"250721","0.62","5.7","176.5","2.5","55.1","3.1","18.7","0.0","70.6","92.3",
"260700","0.71","5.7","171.8","6.8","56.2","8.4","19.0","0.0","58.3","91.3",
"260703","0.75","5.7","169.4","11.1","42.3","13.8","19.6","0.0","45.6","88.7",
"260706","0.78","5.7","166.3","9.3","46.0","11.8","19.5","0.0","47.2","86.5",
"260709","0.78","5.7","162.7","4.9","33.1","6.2","20.5","0.0","59.9","86.7",
"260712","0.76","5.7","158.9","4.3","59.0","5.5","21.3","0.0","48.8","81.7",
"260715","0.74","5.7","156.0","8.0","47.8","10.4","23.0","0.0","10.7","66.8",
"260718","0.75","5.7","154.2","11.1","46.3","14.2","21.2","0.0","6.7","68.2",
"260721","0.77","5.7","153.4","15.3","35.7","19.5","21.1","0.0","26.2","75.2",
"270700","0.79","6.5","153.0","13.9","37.0","18.0","21.2","0.0","34.5","78.7",
"270703","0.80","6.5","152.9","11.9","39.3","15.7","20.9","0.0","0.0","77.0",
"270706","0.77","6.5","153.3","10.2","41.2","13.3","20.9","0.0","0.0","70.1",
"270709","0.74","5.7","152.8","8.1","45.5","10.4","22.8","0.0","0.0","62.6",
"270712","0.70","5.7","151.2","6.0","50.9","7.8","23.8","0.0","0.0","62.1",
"270715","0.63","5.7","149.2","5.5","50.2","7.2","24.4","0.0","0.0","60.8",
"270718","0.58","5.7","147.3","9.9","41.2","13.1","21.8","0.0","0.0","66.6",
"270721","0.57","5.7","145.8","13.6","39.1","17.9","22.1","0.0","0.0","68.1",
"280700","0.57","5.7","145.1","12.8","40.5","16.5","22.5","0.0","0.0","67.4",
"280703","0.57","5.7","144.8","9.8","43.8","12.6","22.0","0.0","0.0","65.9",
"280706","0.56","5.7","144.4","11.1","39.7","14.1","22.1","0.0","0.0","60.8",
"280709","0.54","5.7","143.6","8.8","40.0","11.2","24.1","0.0","0.0","54.8",
"280712","0.49","5.7","142.2","7.1","46.0","9.1","24.9","0.0","0.0","55.1",
"280715","0.46","5.7","140.8","6.1","46.1","8.0","24.9","0.0","0.0","55.7",
"280718","0.43","5.7","139.6","10.1","45.4","13.3","22.9","0.0","0.0","57.4",
"280721","0.40","5.7","138.9","12.8","44.9","17.0","23.6","0.0","17.1","56.3",
"290700","0.38","5.7","138.2","12.9","40.3","16.6","24.0","0.0","26.6","55.4",
"290703","0.37","5.7","137.3","10.0","28.1","12.6","23.2","0.0","22.6","49.7",
];

var CtrlData = [ "Time","Hsto","Tpto","Dmto","Vn10","Dv10","Raj",
"Temp","Prec","Nuv","Umid","","(m)","(s)","(°)","(nos)","(°)","(nos)","(°)","(mm)","(%)","(%)","","1",
,"7","","10","","12","","0.5","","","2","10","","15","","18","","1","",""]];

var HeadersData = [ "Time","Hsto","Tpto","Dmto","Hmva","Vn10","Dv10","Raj",
"Temp","Prec","Nuv","Umid" ];

var UnitsData = [ "(ddmmhh)","(m)","(s)","(°)","(nos)","(°)","(nos)","(°)","(mm)","(%)","(%)" ];

var TipsData = [ "Data da medida","Altura significativa ondas","Período de pico ondas","Direção
média ondas","Intensidade de ventos a 10m","Dirção de ventos a 10m","Rajadas de ventos a 10m",
"Temperatura do ar","Precipitação Acumulada","Cobertura de Nuvens","Umidade Relativa do Ar" ];

var MAXCOLS = 11;

=====
```


ANEXO 3

EXECUTOR_OPERA.SH

```
=====

#!/bin/bash
#####
# N E W   E X E C U T O R _ O P E R A   -   S I P S   -   S I A A   #
# Felipe Chagas e Alex Luz                               22/08/2018 #
#####
#
# Executa '.cronfile' para carregar variaveis NETCDF para o WW3
. $HOME/.cronfile > /dev/null 2>&1
#
# Diretorio Home do user OPERA e arquivo de LOG
OPER=/home/opera
LOG=$OPER/rooms/engine/automata/logs/executor_opera_log.txt
LOGD=$OPER/rooms/engine/automata/logs/executor_opera_DETAILS_log.txt
echo "*-----*" > $LOGD
#### Limpa log DETAILS a cada execucao do executor ####
#
#
reporter()                ##### Function to report errors via MAIL ###
{
MSG="$1"
echo "$MSG" | mail -r pontos@siaabr.com -v -s "ERRO EXECUTOR_OPERA PONTOS" operational@siaabr.com
alexluz@siaabr.com >> $LOGD 2>&1
}
#
#
echo "*-----*" >> $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) Inicio do Executor OPERA" >> $LOG
#
### Folder com rotinas e softwares de automatizacao
AUTOM=$OPER/rooms/engine/automata >> $LOGD 2>&1
#
### Data atual do sistema
ANO="(date +%Y)"
MES="(date +%m)"
DIA="(date +%d)"
#
#### 1.0 Nivel Global
FOLGLOB=$OPER/rooms/suits/lglobal >> $LOGD 2>&1
#
### Cria arquivo wget_GFS_global para baixar VENTOS globais ###
octave --no-gui --silent $FOLGLOB/automata/cria_wget_GFS_global.m >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Criado arquivo wget_GFS_global com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA CRIACAO DO ARQUIVO WGET_gfs_GLOBAL - EXECUTOR OPERA
TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
#
### Baixa VENTOS globais ###
chmod -R g+rxw $FOLGLOB/data/temp/wget_GFS_global >> $LOGD 2>&1
$FOLGLOB/data/temp/wget_GFS_global >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Baixado GFS Global com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO AO BAIXAR GFS GLOBAL - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
#
###Chama WINDER2 para converter VENTOS para formato WW3 ###
###$FOLGLOB/data/temp/winder2 "$FOLGLOB/data/temp/gfs_0p50_t06z.pgrb2full_$(($ANO))$(($MES))$(($DIA))06_f*"
"UGRD,VGRD" >> $LOG ### Deixar 10m above ###
$AUTOM/winder2 "$FOLGLOB/data/temp/gfs_0p50_t06z.pgrb2full_$(($ANO))$(($MES))$(($DIA))06_f*" "UGRD,VGRD" >>
$LOGD 2>&1 ### Deixar 10m above ###
```

```

#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Winder2 executado com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO WINDER2 - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
### Copia o arquivo com os componentes U e V gerado pelo WINDER2 para modelo WW3 processar ###
cp $FOLGLOB/data/latest_gfs/latest_gfs.dat $OPER/demo/ww3/inpvar/ >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Copiado latest_gfs.dat com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA COPIA DO LATEST_GFS.DAT - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
### BACKUP do arquivos UV de VENTOS ###
cp $FOLGLOB/data/latest_gfs/latest_gfs.dat $FOLGLOB/data/backup_gfs/ >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Backupeado latest_gfs.dat com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NO BACKUP DO LATEST_GFS.DAT - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
### Executa o TESTNODES para checar os nodes antes do WW3 ###
$AUTOM/testnodes >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Testnodes executado com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO TESTNODES - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
### R o d a W W 3 ###
$OPER/demo/ww3/ww3glats_felipe >> $LOGD 2>&1
#
RCWW3=$?
if [ $RCWW3 = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) WW3 Global_At1s executado com sucesso" >> $LOG
else
### echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO WW3 GLOBAL_ATLS - EXECUTOR OPERA TERMINADO"
>> $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO WW3 GLOBAL_ATLS - ***VAI AGUARDAR 5 MINUTOS E
TENTAR NOVAMENTE***" >> $LOG
echo "$(date +%d/%m/%y_%H:%M:%S) ***VAI RODAR TESTNODES NOVAMENTE E WW3 NOVAMENTE SOMENTE MAIS UMA
VEZ***" >> $LOG
MSG=`tail -2 $LOG`; reporter "$MSG"
### exit 1
sleep 5m
fi
#
#####
#####
# SE O WW3 DER ERRO EXECUTA TESTNODES NOVAMENTE E WW3 NOVAMENTE - I N I C I O
#####
#####
if [ $RCWW3 != 0 ]
then
### Executa o TESTNODES para checar os nodes antes do WW3 ###
$AUTOM/testnodes >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Testnodes executado com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO TESTNODES - EXECUTOR OPERA TERMINADO" >> $LOG

```

```

MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
### R o d a   W W 3 ###
$OPER/demo/ww3/ww3glats_felipe >> $LOGD 2>&1
#
RC=$?
if [ $RC = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) WW3 Global_AtlS executado com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA EXECUCAO DO WW3 GLOBAL_ATLS - EXECUTOR OPERA TERMINADO" >>
$LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
fi
#####
#####
# SE O WW3 DER ERRO EXECUTA TESTNODES NOVAMENTE E WW3 NOVAMENTE -           F I M
#####
#####
#
### Copia resultados de WW3 para Nautilus
scp $FOLGLOB/data/latest_ww3/gridatls09min.nc
nemo@nautilus:rooms/suits/lglobal/latlas/data/latest_ww3 >> $LOGD 2>&1
RC09=$?
scp $FOLGLOB/data/latest_ww3/gridglo30min.nc nemo@nautilus:rooms/suits/lglobal/data/latest_ww3 >>
$LOGD 2>&1
RC30=$?
#
if [ $RC09 = 0 -a $RC30 = 0 ]
then
echo "$(date +%d/%m/%y_%H:%M:%S) Resultados .NCs copiados para o NAUTILUS com sucesso" >> $LOG
else
echo "$(date +%d/%m/%y_%H:%M:%S) ERRO NA COPIA DOS RESULTADOS .NCs PARA O NAUTILUS - RC09 = $RC09 -
RC30 = $RC30 - EXECUTOR OPERA TERMINADO" >> $LOG
MSG=`tail -1 $LOG`; reporter "$MSG"
exit 1
fi
#
#
#
##### 1.1 Nivel Atlântico Sul
#
##### 1.1.1 Nivel SP-RJ
#
##### 1.1.1.1 Nivel SS
#
#
#
echo "$(date +%d/%m/%y_%H:%M:%S) F I M do Executor OPERA" >> $LOG
#
#
#

```

=====

TESTNODES

=====

```

#!/bin/bash
#####
# TESTA OS NODES DO PONTOS ANTES DE RODAR WW3 #
# REcria O MACHINEFILE CONFORME OS NODES ATIVOS E INATIVOS #
# Alex Luz - SIAA - 06/08/2018 #
#####
#
echo "$(date +%d/%m/%y_%H:%M:%S) INICIO do testnodes"
echo
#
MF=/home/opera/demo/ww3/machinefile
PW=/home/opera/demo/ww3/procs_ww3
#
cp $MF ${MF}_OLD; cat $MF; echo
#
### Pega os nodes do /etc/hosts exceto o node00 e joga no array NODES

```

```

NODES=$(egrep -i node /etc/hosts | egrep -v node00 | awk {'print $2'})
echo ${NODES[*]}; echo ${#NODES[*]}; echo
#
declare -i TOT=0 TNA=0 TNI=0 TPA=0 DIF=0 CON=0
declare -i MIN=4 MAX=8; TOT=${#NODES[*]}          ##### Define MIN e MAX e TOT nodes ativos
#####
#
for NOD in ${NODES[*]}
do
#
declare -i ST$NOD=1
ping -c1 $NOD|egrep -i Unreachable > /dev/null 2>&1 && let ST$NOD=1 || let ST$NOD=0          #####
Status do ping de cada node = 0 (OK) ou 1 (NOK - Unreachable)
echo "$NOD" = $(eval echo \$$ST$NOD)
TNI=$TNI+$(eval echo \$$ST$NOD)
#
done
#
TNA=$(expr $TOT - $TNI)
#
echo
echo "Total de nodes ATIVOS =" $TNA
echo "Total de nodes INATIVOS =" $TNI
echo
#
[ $TNA -lt $MIN ] && { echo "ATENCAO! TOTAL NODES ATIVOS MENOR QUE $MIN. ***TESTNODES TERMINADO COM
RC=1***"; echo; exit 1; }
#
if [ $TNA -gt $MAX ]
then
echo "ATENCAO! TOTAL NODES ATIVOS (TNA) MAIOR QUE $MAX. TESTNODES CONTINUA COM RC=0 ***MAS ALTERA
TNA PARA $MAX***"
#
DIF=$TNA-$MAX
TNA=$MAX
TNI=$TNI+$DIF
#
echo
echo "NOVO total de nodes ATIVOS =" $TNA
echo "NOVO total de nodes INATIVOS =" $TNI
#
fi
#
TPA=$(expr $TNA \* 8)
echo
echo "Total de processadores ATIVOS =" $TPA
echo
#
echo $TPA > $PW
#
rm -f $MF; touch $MF
#
CON=0
for NOD in ${NODES[*]}
do
#
[ $(eval echo \$$ST$NOD) -eq 0 -a $CON -lt $MAX ] && echo "$NOD:$TNA" >> $MF || echo "$NOD:$TNA" >>
$MF          ##### Recria o machinefile conforme os status de cada node #####
[ $(eval echo \$$ST$NOD) -eq 0 ] && let CON++
##### e conforme a qtde MAX de nodes permitidos.          #####
#
##### Status 0 'node01:8' - Status 1 '#node01:8' - EXEMPLO #####
done
#
#
cat $MF
echo
cat $PW
#
echo
echo "$(date +%d/%m/%y_%H:%M:%S) FIM normal do testnodes"
#
#
#
=====

```

=====

53

```

done                                     ##### done for DIR #####
#
if [ "$QWPS" != "$QWPW" ]
then
MSG1="###QTDES### de arquivos diferentes nos diretorios WEBPOINTS_SWAN_WRF $CLIENT $DIR"
MSG2="$(date +%d/%m/%y_%H:%M:%S) EXEC_CLIENT ENCERRADO COM RC=1"
echo $MSG1; echo $MSG2
reporter "$MSG1 $MSG2"
exit 1
fi
#
diff ${BASE_DIR}NwpS`basename ${CLIENT}` ${BASE_DIR}NwpW`basename ${CLIENT}` 2> /dev/null
RC=$?
#
if [ $RC = 1 ]
then
MSG1="###NOMES### de arquivos diferentes nos diretorios WEBPOINTS_SWAN_WRF $CLIENT $DIR"
MSG2="$(date +%d/%m/%y_%H:%M:%S) EXEC_CLIENT ENCERRADO COM RC=1"
echo $MSG1; echo $MSG2
reporter "$MSG1 $MSG2"
exit 1
fi
#
diff ${BASE_DIR}LwpS`basename ${CLIENT}` ${BASE_DIR}LwpW`basename ${CLIENT}` 2> /dev/null
RC=$?
#
if [ $RC = 1 ]
then
MSG1="###LINHAS### de arquivos diferentes nos diretorios WEBPOINTS_SWAN_WRF $CLIENT $DIR"
MSG2="$(date +%d/%m/%y_%H:%M:%S) EXEC_CLIENT ENCERRADO COM RC=1"
echo $MSG1; echo $MSG2
reporter "$MSG1 $MSG2"
exit 1
fi
#
diff ${BASE_DIR}DwpS`basename ${CLIENT}` ${BASE_DIR}DwpW`basename ${CLIENT}` 2> /dev/null
RC=$?
#
if [ $RC = 1 ]
then
MSG1="###DATAS### de arquivos diferentes nos diretorios WEBPOINTS_SWAN_WRF $CLIENT $DIR"
MSG2="$(date +%d/%m/%y_%H:%M:%S) EXEC_CLIENT ENCERRADO COM RC=1"
echo $MSG1; echo $MSG2
reporter "$MSG1 $MSG2"
exit 1
fi
#
done                                     ##### done for CLIENT #####
#
echo "$(date +%d/%m/%y_%H:%M:%S) F i m da checagem"
#
}
#
#
convert()                               ##### Converte CSV em report.js #####
{
#
cd $BASE_DIR
#
FILE_C="$1";
OUTPUT="$2";
#
REPORT_JS="report_`basename ${FILE_C}`";
TEMP=$(mktemp);
#
test -s "$FILE_C" || { echo "ARQUIVO $FILE_C INVALIDO...!"; continue;}
#
test -z "$OUTPUT" && OUTPUT="$REPORT_JS"
#
sed "s/^/\"/g" $FILE_C | sed "s/\",/\"\\\",/g" | sed "s/$/\"\\\",/g" > $TEMP
#
eval $(eval egrep -i HeadersData $REPORT_JS_BASE | sed "s/[/\\(/g" | sed "s/[/\\)/g" | sed "s/ //g" |
sed "s/\",/\" /g" | sed "s/^var//g")      ##### Cria array com os HeadersData #####
#
I=1
for COL in ${HeadersData[*]}
do
#
cat $TEMP | cut -d"," -f ${I} > $COL
let I++
#
done
#
AUX_C1=$(mktemp)
AUX_C2=$(mktemp)
for FILE_COL in ${HeadersData[*]}
do
#

```

```

paste -d"," $AUX_C1 $FILE_COL > $AUX_C2
cp $AUX_C2 $AUX_C1
#
done
#
sed "s/^,//g" $AUX_C1 | sed "s/$/,/g" > $TEMP
#
###paste -d"," timeCS hstoC tptoC dmtoc vn10C dv10C rajC tempC precC nuvc humC | sed "s/$/,/g" >
$TEMP
#
FLAD="var Data = [ "; echo $FLAD > fladR
LLAD="];"; echo $LLAD > lladR
#
cat fladR $TEMP lladR $REPORT_JS_BASE > ${CLIENT}/webtabs/$REPORT_JS
#
for FILE_COL in ${HeadersData[*]}; do rm -f $FILE_COL; done
#
rm -f $TEMP $AUX_C1 $AUX_C2 fladR lladR
#
###echo ${CLIENT}/webtabs/$REPORT_JS
#
}
#
#
join()          ##### Junta os CSVs originais e os converte (convert function) para os reports.js
#####
#          ##### Cria os CSVs para enviar em anexos aos e-mails aos clientes #####
{
#
AUX_J=$(mktemp)
MAXCOLS=$(egrep -i MAXCOLS $REPORT_JS_BASE | cut -d" " -f4 | sed "s/;/,/g")
#
###echo ${ACLIS[*]} ARRAY ACLIS JOIN...
#
for CLIENT in ${ACLIS[*]}
do
#
for FILE_J in `cat ${BASE_DIR}NwpS\`basename ${CLIENT}\` 2> /dev/null`
do
#
cut -d"," -f2-${MAXCOLS} ${CLIENT}/webpoints_wrf/${FILE_J} > $AUX_J          ##### Tira a PRIMEIRA
coluna (DATA) do arquivo CSV webpoints_wrf - Cols Datas sao iguais #####
#
### paste -d"," ${CLIENT}/webpoints_swan/${FILE_J} ${CLIENT}/webpoints_wrf/${FILE_J} >
${CLIENT}/webtabs/${FILE_J%.*}.js
paste -d"," ${CLIENT}/webpoints_swan/${FILE_J} $AUX_J > ${CLIENT}/webtabs/${FILE_J%.*}.js
#
convert ${CLIENT}/webtabs/${FILE_J%.*}.js && rm ${CLIENT}/webtabs/${FILE_J%.*}.js
##### Funcao 'convert' acima #####
#
egrep ^\` ${CLIENT}/webtabs/report_${FILE_J%.*}.js > ${CLIENT}/webtabs/report_${FILE_J%.*}.csv
##### Cria os CSVs para enviar os e-mails aos clientes #####
#
### echo ${CLIENT}/webtabs/report_${FILE_J%.*}.csv
#
done
#
done
#
rm -f ${BASE_DIR}NwpS* ${BASE_DIR}NwpW*
rm -f ${BASE_DIR}LwpS* ${BASE_DIR}LwpW*
rm -f ${BASE_DIR}DwpS* ${BASE_DIR}DwpW*
#
echo "$(date +%d/%m/%Y_%H:%M:%S) reports.js e reports.csv criados"
#
}
#
#
transfer()          ##### Copia reportS.js e JSONs para AWS (Cliente 001public <=> Pasta 'public'
da AWS) #####
#          ##### Copia reportS.js e JSONs para AWS (Cliente 002ss <=> Pasta 'ss'
da AWS) #####
#          ##### Copia reportS.js e JSONs para AWS (Cliente 003demo <=> Pasta 'demo'
da AWS) #####
{
#
###echo ${ACLIS[*]} ARRAY ACLIS TRANSFER...
#
for CLIENT in ${ACLIS[*]}
do
#
DIR_CLIENT=$(basename $CLIENT|sed "s/[0-9]//g")
#
scp ${CLIENT}/webtabs/*.js ec2-user@modisiaabr.com:/var/www/html/${DIR_CLIENT}/code/reports/
echo "scp ${CLIENT}/webtabs/*.js ec2-user@modisiaabr.com:/var/www/html/${DIR_CLIENT}/code/reports/"
#
scp ${CLIENT}/webgrids_swan/*.json ec2-user@modisiaabr.com:/var/www/html/${DIR_CLIENT}/data/wave/

```

```

echo "scp ${CLIENT}/webgrids_swan/*json ec2-
user@modsiaabr.com:/var/www/html/${DIR_CLIENT}/data/wave/"
scp ${CLIENT}/webgrids_wrf/*json ec2-user@modsiaabr.com:/var/www/html/${DIR_CLIENT}/data/wind/
echo "scp ${CLIENT}/webgrids_wrf/*json ec2-
user@modsiaabr.com:/var/www/html/${DIR_CLIENT}/data/wind/"
#
scp ${CLIENT}/webtabs/report_entradaSS.js ec2-
user@modsiaabr.com:/var/www/html/${DIR_CLIENT}/code/report.js ##### QG copia 'report_entradaSS.js'
PUBLIC/SS/DEMO para 'report.js #####
echo "scp ${CLIENT}/webtabs/report_entradaSS.js ec2-
user@modsiaabr.com:/var/www/html/${DIR_CLIENT}/code/report.js"
#
done
#
echo "$(date +%d/%m/%y_%H:%M:%S) reports.js e JSONs copiados para AWS..."
#
}
#
#
sendcsv()
{
#
EMAIL_CLIENT=$1
EMAIL_SIAA=$2
SUBJECT=$3
DIR_CLIENT=$4
BODY=$5
ATTACHED=/home/nemo/rooms/steaks/${DIR_CLIENT}/webtabs/report*csv
#
echo "$BODY" | mutt -a $ATTACHED -s "$SUBJECT" -b alexluz@siaabr.com -c "$EMAIL_SIAA" --
"$EMAIL_CLIENT" ##### TIRAR O MEU E-MAIL... #####
#
echo "$(date +%d/%m/%y_%H:%M:%S) e-mail com reports.CSV anexos enviado ao cliente ${DIR_CLIENT}"
#
}
#
#
main ()
{
##### MAIN #####
{
#
check
#
join
#
transfer
#
}
#
#
main
#
#
# EMAIL_CLIENT $1 EMAIL_SIAA $2 SUBJECT $3
DIR-CLI $4 BODY OF EMAIL... $5
Visualize também no sistema web: http://ss.modsiaabr.com/tabler.html"dia $(date +%d/%m/%y) - Canal de
São Sebastião" "002ss" "Reports do dia $(date +%d/%m/%y) em anexo.
#
EMAIL_CLIENT $1 EMAIL_SIAA $2 SUBJECT $3
DIR-CLI $4 BODY OF EMAIL... $5
Visualize também no sistema web: http://demo.modsiaabr.com/tabler.html"a $(date +%d/%m/%y) - Canal do
Rio de Janeiro" "003demo" "Reports do dia $(date +%d/%m/%y) em anexo.
#
EMAIL_CLIENT $1 EMAIL_SIAA $2 SUBJECT $3
DIR-CLI $4 BODY OF EMAIL... $5
Visualize também no sistema web: http://public.modsiaabr.com/tabler.html"$(date +%d/%m/%y) - Porto do
Cliente Public" "001public" "Reports do dia $(date +%d/%m/%y) em anexo.
#
#
echo "$(date +%d/%m/%y_%H:%M:%S) F I M do Executor CLIENT"
#
#
#
=====

```